# Machine Learning

GNU emacser

# 차 례

# 1 소개

## 1.1 기계 학습이란

### 1.1.1 기계 학습의 정의

### 1.1.2 지식기반 방식에서 기계 학습으로의 대전환

### 1.1.3 기계 학습 개념

### 1.1.4 사람의 학습과 기계 학습

## 1.2 특징 공간에 대한 이해

### 1.2.1 1차원과 2차원 특징 공간

### 1.2.2 다차원 특징 공간

### 1.2.3 특징 공간 변환과 표현 학습

## 1.3 데이터에 대한 이해

### 1.3.1 데이터 생성 과정

### 1.3.2 데이터베이스의 중요성

### 1.3.3 데이터베이스 크기와 기계 학습 성능

### 1.3.4 데이터 가시화

## 1.4 간단한 기계 학습의 예

## 1.5 모델 선택

### 1.5.1 과소적합과 과잉적합

### 1.5.2 바이어스와 분산

### 1.5.3 검증집합과 교차검증을 이용한 모델 선택 알고리즘

### 1.5.4 모델 선택 한계와 현실적인 해결책

## 1.6 규제

### 1.6.1 데이터 확대

### 1.6.2 가중치 감쇄

## 1.7 기계 학습 유형

### 1.7.1 지도 방식에 따른 유형

### 1.7.2 다양한 기준에 따른 유형

## 1.8 기계 학습의 과거와 현재, 미래

### 1.8.1 인공지능과 기계 학습의 간략한 역사

### 1.8.2 기술 추세

### 1.8.3 사회적 전망

# 2 기계 학습과 수학

## 2.1 선형대수

### 2.1.1 벡터와 행렬

#### 행렬 연산

#### 텐서

### 2.1.2 놈과 유사도

#### 놈

벡터의 크기를 정의할 때는 놈을 사용한다. 2차 놈 $\|x\|_2$를 유클리디언 놈 또는 L2놈이라 하고, 첨자를 생략하고 $\|x\|$로 표기할 수도 있다.
$p = \infty$일 때의 놈을 최대 놈이라 부르고, 식(2.4)로 정의한다.

$$p\text{차 놈: } \|x\|_p = \left( \sum_{i=1,d} |x_i|^p \right)^{\frac{1}{p}} \tag{2.3}$$

$$\|x\|_\infty = \max(|x_1|, |x_2|, \cdots, |x_d|) \tag{2.4}$$

$$\text{단위 벡터: } \frac{x}{\|x\|_2} \tag{2.5}$$

$$\text{프로베니우스 놈: } \|A\|_F = \left( \sum_{i=1,n} \sum_{j=1,m} a_{ij}^2 \right) \tag{2.6}$$

#### 유사도와 거리

$$cosine\_similarity(a, b) = \frac{a}{\|a\|} \cdot \frac{b}{\|b\|} = \cos(\theta) \tag{2.7}$$

### 2.1.3 퍼셉트론의 해석

#### 학습의 정의

### 2.1.4 선형결합과 벡터공간

#### 선형결합이 만드는 벡터공간

#### 기계 학습의 공간 변환

### 2.1.5 역행렬

**행렬식**

**정부호 행렬**

양의 정부호 행렬: $0$이 아닌 모든 벡터$x$에 대해, $\mathbf{x}^T\mathbf{A}\mathbf{x} > 0$ (2.18 + 2.19)
양의 준정부호 행렬: $0$이 아닌 모든 벡터$x$에 대해, $\mathbf{x}^T\mathbf{A}\mathbf{x} \geq 0$
음의 정부호 행렬: $0$이 아닌 모든 벡터$x$에 대해, $\mathbf{x}^T\mathbf{A}\mathbf{x} < 0$
음의 준정부호 행렬: $0$이 아닌 모든 벡터$x$에 대해, $\mathbf{x}^T\mathbf{A}\mathbf{x} \leq 0$

### 2.1.6 행렬 분해

**고윳값과 고유 벡터**
식 (2.20)을 만족하는 $(0)$이 아닌 벡터 $\mathbf{v}$를 행렬 $\mathbf{A}$의 eigen vector 라고 하며 eigen vector에 대응하는 실수 $\lambda$를 eigen value라 한다.

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v} \tag{2.20}$$

모든 고유 벡터는 서로 직교이다.

**고윳값 분해**
eigen value decomposition은 행렬 $A$를 식 (2.21)과 같이 분해한다.

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q} \tag{2.21}$$

$\Lambda$는 교윳값을 대각선에 배치한 대각행렬이고, $Q$는 행 벡터나 열 벡터가 서로 직교인 직교행렬이다.
EVD는 정사각행렬이 아니면 적용할 수 없다. 정사각행렬이 아닌 경우에는 SVD가 쓰인다.

**특잇값 분해**
SVD는 어떤 행렬이든 적용할 수 있다.

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathbf{T} \tag{2.22}$$

$U$는 $\mathbf{A}\mathbf{A}^T$의 고유 벡터를 열에 배치한 $n * n$ 크기의 왼쪽 특이 행렬이다.
$V$는 $\mathbf{A}\mathbf{A}^T$의 고유 벡터를 열에 배치한 $m * m$ 크기의 오른쪽 특이 행렬이다.
$\Sigma$는 $\mathbf{A}\mathbf{A}^T$의 고윳값의 제곱근을 대각선에 배치한 $n * m$크기의 대각행렬이다.

## 2.2 확률과 통계

### 2.2.1 확률 기초

**확률변수와 확률분포**

**곱 규칙과 합 규칙**

### 2.2.2 베이즈 정리와 기계 학습

**베이즈 정리**

**기계 학습에 적용**

### 2.2.3 최대 우도

**최대 우도법**

**기계 학습에 적용**

### 2.2.4 평균과 분산

**평균 벡터와 공분산 행렬**

### 2.2.5 유용한 확률분포

#### 가우시안 분포

가우시안 분포는 평균과 분산을 나타내는 2개의 매개변수 $\mu$와 $\sigma^2$으로 규정하며, 식 (2.40)으로 정의한다. 정규분포라고도 하며 $\mathsf{N}(x;\mu,\sigma^2)$과 같이 표기한다. $\mu$에서 최대값을 가지고, $\sigma^2$는 분포의 퍼진 정도를 나타내는데, $\sigma^2$값이 클수록 봉우리의 높이가 낮고 좌우로 멀리 퍼진다.

$$\mathsf{N}(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \tag{2.40}$$

특징 벡터가 $d$차원인 가우시안 분포는 평균 벡터 $\mu$와 공분산 행렬 $\Sigma$라는 매개변수로 모양이 규정된다. 식 (2.41)은 다차원 가우시안 분포이다. $|\Sigma|$은 $\Sigma$의 행렬식, 즉 $\det(\Sigma)$이다.

$$\mathsf{N}(x;\mu,\Sigma^{\mathbf{2}}) = \frac{1}{\sqrt{|\Sigma|}\sqrt{(2\pi)^d}}\exp(-\frac{1}{2}\left(\mathsf{x}-\mu\right)^T\Sigma^{-1}(\mathsf{x}-\mu)\big) \tag{2.41}$$

#### 베르누이 분포와 이항 분포

확률변수 $x$가 1 또는 0의 두가지 값만 가질 수 있는 이진변수이고, 성공확률은 $p$, 실패확률은 $1-p$인 분포를 베르누이 분포라고 한다. 매개변수는 $p$ 하나이다.

$$Ber(x;p) = p^x(1-p)^{1-x} = \begin{cases} p, \ x=1\text{일 때} \\ 1-p, \ x=0\text{일 때} \end{cases} \tag{2.42}$$

성공확률이 $p$인 베르누이 실험을 $m$번 수행할 때 성공할 회수의 확률분포를 이항분포라고 한다. 매개변수는 $p$와 $m$으로 2개이다.

$$B(x;m,p) = C_m^x p^x(1-p)^{m-x} = \frac{m!}{x!(m-x)!}p^x(1-p)^{m-x} \tag{2.43}$$

### 2.2.6 정보이론

#### 자기 정보와 엔트로피

사건 $e_i$의 정보량 $h(e_i)$를 자기 정보라 한다.

$$h(e_i) = -\log_2 P(e_i)\text{또는}h(e_i) = -\log_e P(e_i) \tag{2.44}$$

밑이 2면 단위는 bit이고, $e$면 나츠nat이다.

$$\text{이산확률분포} \quad H(x) = -\sum_{i=1,k} P(e_i)\log_{2\,or\,e} P(e_i) \tag{2.45}$$

$$\text{연속확률분포} \quad -\int_{\mathbb{R}} P(x)log_{2\,or\,e}P(e_i) \tag{2.46}$$

**교차 엔트로피와 KL 다이버전스**

식 (2.47)은 $P$와 $Q$사이의 교차 엔트로피를 정의한다. 이때 두 확률분포는 같은 확률변수에 의해 정의되어 있어야 한다.

$$H(P,Q) = -\sum_{x} P(x)\log_2 Q(x) = -\sum_{i=1,k} P(e_i)\log_2 Q(e_i) \tag{2.47}$$

식 (2.47)의 유도

$$\begin{aligned} H(P,Q) &= -\sum_{x} P(x)\log_2 P(x) \\ &= -\sum_{x} P(x)\log_2 P(x) + \sum_{x} P(x)\log_2 P(x) - \sum xP(x)\log_2 Q(x) \\ &= H(P) + \sum_{x} P(x)\log_2 \frac{P(x)}{Q(x)} \end{aligned}$$

$$KL(P\|Q) = \sum_{x} P(x) log_2 \frac{P(x)}{Q(x)} \tag{2.48}$$

$$P\text{와 } Q\text{의 교차 엔트로피 } H(P,Q) = H(P) + \sum_{x} \log_2 \frac{P(x)}{Q(x)} \tag{2.49}$$

$$= P\text{의 엔트로피 } + P\text{와 } Q\text{간의 } KL\text{다이버전스} \tag{1}$$

KL 다이버전스는 두 확률분포가 얼마나 다른지 측정한다. 거리 개념을 내포하지만 $KL(P\|Q) \neq KL(Q\|P)$ 이므로 엄밀한 수학적 정의에 따르면 거리가 아니다.

## 2.3 최적화

### 2.3.1 매개변수 공간의 탐색

**학습 모델의 매개변수 공간**

**최적화 문제해결**

### 2.3.2 미분

**미분에 의한 최적화**

**편미분**

**독립변수와 종속변수의 구분**

**연쇄법칙**

**야코비안과 헤시안**

**테일러 급수**

$$f(x+\Delta) = f(x) + f'(x)\Delta + \frac{f''(x)}{2}\Delta^2 + \cdots + \frac{f^{(k)}(x)}{k!}\Delta^k + \cdots \approx f(x) + f'(x)\Delta \qquad (2.57)$$

### 2.3.3 경사 하강 알고리즘

# 3 다층 퍼셉트론

## 3.1 신경망 기초

### 3.1.1 인공신경망과 생물신경망

### 3.1.2 신경망의 간략한 역사

### 3.1.3 신경망의 종류

## 3.2 퍼셉트론

### 3.2.1 구조

### 3.2.2 동작

$$y = \tau(s)$$

$$\text{이때 } s = w_0 + \sum_{i=1}^{d} w_i x_i, \qquad \tau(s) = \begin{cases} 1 & s \geq 0 \\ -1 & s < 0 \end{cases} \tag{2}$$

**행렬 표기**

**분류기로 해석**

### 3.2.3 학습

**목적함수 설계와 델타 규칙 유도**

$$J(\mathbf{w}) = \sum_{\mathbf{x}_k \in Y} -y_k(\mathbf{w}^T \mathbf{x}_k) \tag{3.7}$$

$$\text{델타 규칙: } w_i = w_i + \rho \sum_{\mathbf{x}_k \in Y} y_k x_{ki}, \qquad i = 0, 1, \cdots, d \tag{3.9}$$

**학습 알고리즘**

**행렬 표기**

## 3.3 다층 퍼셉트론

### 3.3.1 특징 공간 변환

**다층 퍼셉트론의 용량**

### 3.3.2 활성함수

Rectifier는 gradient 소멸 문제를 크게 완화하고, 속도 향상에 유용하다.
Rectifier의 음수를 0으로 대치하는 문제를 해결하기 위한 여러 변종이 개발되어 있다.

### 3.3.3 구조

**병렬분산 구조**

### 3.3.4 동작

행렬을 사용하면 2층 퍼셉트론의 동작을 식 (3.15)와 같이 간결하게 표기할 수 있다.

$$\mathbf{o} = \tau(\mathbf{U}^2\tau_h(\mathbf{U}^1\mathbf{x})) \tag{3.15}$$

**은닉층은 특징 추출기**
은닉층은 원래 특징 공간을 새로운 특징 공간으로 변환하는 특징 추출기로 볼 수 있다.
계층적 방식은 앞 쪽에 있는 은닉층들이 추출한 저급 특징에서 뒤쪽에있는 은닉층들이 고급 특징을 추출하는 방식이다.

## 3.4 오류 역전파 알고리즘

### 3.4.1 목적함수의 정의

훈련집합이 $\mathbb{X} = \{x_1, x_2, \cdots, x_n\}, \mathbb{Y} = \{y_1, y_2, \cdots, y_n\}$ 이라 가정하면, $x_i$는 $i$번째 샘플의 특징 벡터이고 $y_i$는 소속 부류이다. 소속 부류가 $j$번째이면 $y_i = (0, 0, \cdots, 1, 0, \cdots, 0)^T$ (j번째 요소만 1이고 나머지는 전부 0인 벡터) 이다.
행렬을 사용하면 훈련집합 $\mathbb{X}$와 $\mathbb{Y}$를 식 (3.16)처럼 특징 벡터 행렬 $X$와 소속 부류 행렬 $Y$로 표기할 수 있다. $X$와 $Y$는 각각 $n*d$와 $n*c$ 행렬이다. $n$은 샘플의 개수, $d$는 특징 벡터의 차원, $c$는 부류의 개수이다.

$$X = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix}, \quad Y = \begin{pmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_n^T \end{pmatrix} \tag{3.16}$$

기계 학습의 궁극적인 목적은 식 (3.17)과 같이 $X$를 완벽하게 $Y$로 매핑하는 최적의 함수 $F$를 알아내는 것이다. 다시 말해 모든 샘플을 옳게 분류하는 분류기인 $f$를 찾아야 하지만, 완벽한 분류기는 불가능하므로 근사 최적해를 구한다.

$$\left.\begin{array}{l} Y = f(X) \\ \text{풀어 쓰면} \quad y_i = f(x_i), i = 1, 2, \cdots, n \end{array}\right\} \tag{3.17}$$

식 (3.17)은 $f(X)$로 표기했지만 엄밀하게 표기하면 $f(X;\Theta)$와 같다. $f$가 $\Theta$로 매개변수화되어 있다는 사실을 드러내는 표기이다. 2층 퍼셉트론에서는 $\Theta = \{U^1, U^2\}$이다.
이제 기계 학습이 해야 할 일을 식 (3.18)과 같이 기술할 수 있다.

$$\hat{\Theta} = \underset{\Theta}{\text{argmax}} ||f(X;\Theta) - Y||_2^2 \tag{3.18}$$

다시 말해, 기계 학습은 다층 퍼셉트론의 출력 $f(X;\Theta)$와 주어진 부류 정보 $Y$의 차이를 최소로 하는 최적의 매개변수 $\hat{\Theta}$을 찾아야 한다. 이 사실을 토대로 목적함수를 설계하면 식 (3.19)와 같다.

$$\left.\begin{array}{ll} \text{온라인 모드:} & e = \dfrac{1}{2}||y - o||_2^2 \\ \text{배치 모드:} & e = \dfrac{1}{2n}\displaystyle\sum_{i=1}^{n} ||y_i - o_i||_2^2 \end{array}\right\} \tag{3.19}$$

평균제곱 오차는 주로 다층 퍼셉트론이 사용하는데, 딥러닝은 교차 엔트로피 또는 로그우도를 사용한다. 이 새로운 목적함수은 딥러닝의 성능 향상에 공헌하는 요인이다.

### 3.4.2 오류 역전파 알고리즘 설계

식 (3.19)에 있는 온라인 모드의 수식을 매개변수 Θ가 드러나도록 다시 쓰면 식 (3.20)이 된다.

$$J(\Theta) = \frac{1}{2}||y - \mathsf{o}(\Theta)||_2^2 \tag{3.20}$$

기계 학습 알고리즘은 식 (3.20)의 목적함숫값이 줄어드는 방향으로 Θ, 즉 $\mathsf{U}^1$과 $\mathsf{U}^2$의 값을 수정해야 한다. 경사 하강법 원리에 따르면 식 (3.21)이 성립한다. 이 식에서 $\rho$는 학습률이다. 온라인 모드와 배치 모드 중 선택해 사용할 수 있는데 온라인 모드에 대해 설명한다.

$$\left.\begin{aligned} \mathsf{U}^1 &= \mathsf{U}^1 - \rho\frac{\partial J}{\partial U^1} \\ \mathsf{U}^2 &= \mathsf{U}^2 - \rho\frac{\partial J}{\partial U^2} \end{aligned}\right\} \tag{3.21}$$

**오류 역전파의 유도**

$$\begin{aligned} \frac{\partial J}{\partial u_{kj}^2} &= \frac{\partial(0.5\|\mathbf{y} - \mathbf{o}(\mathbf{U}^1, \mathbf{U}^2)\|_2^2)}{\partial u_{kj}^2} \\ &= \frac{\partial(0.5\sum_{q=1}^{c}(y_q - o_q)^2)}{\partial u_{kj}^2} \\ &= \frac{\partial(0.5(y_k - o_k)^2)}{\partial u_{kj}^2} \\ &= -(y_k - o_k)\frac{\partial o_k}{\partial u_{kj}^2} \\ &= -(y_k - o_k)\frac{\partial \tau(osum_k)}{\partial u_{kj}^2} \\ &= -(y_k - o_k)\tau'(osum_k)\frac{\partial osum_k}{\partial u_{kj}^2} \\ &= -(y_k - o_k)\tau'(osum_k)z_j \end{aligned}$$

$$\delta_k = (y_k - o_k)\tau'(osum_k), \qquad 1 \le k \le c \tag{3.22}$$

$$\frac{\partial J}{\partial u_{kj}^2} = -\delta_k z_j, \qquad 0 \le j \le p, 1 \le k \le c \tag{3.23}$$

$$\eta_j = \tau'(zsum_j)\sum_{q=1}^{c}\delta_{qu_{qj}}^2, \qquad 1 \le j \le p \tag{3.24}$$

$$\frac{\partial J}{\partial u_{ji}^1} = \Delta u_{ji}^1 = -\eta_j x_i, \qquad 0 \le i \le d, 1 \le j \le p \tag{3.25}$$

### 3.4.3 오류 역전파를 이용한 학습 알고리즘

**행렬 표기**

## 3.5 미니배치 스토캐스틱 경사 하강법

현대 기계 학습은 스토캐스틱과 배치의 중간에 해당하는 미니배치 방식을 주로 사용한다. 미니배치에서는 $t$를 보통 수십 수백 정도의 크기로 설정한다.
미니배치에서는 학습이 완료될 때까지 한번도 학습에 참여하지 않는 샘플이 있을 수 있지만, 성능에 해를 끼치지는 않는다.
미니배치를 훈련집합을 모두 사용하도록 구현할 수도 있다.
미니배치의 높은 무작위성은 일종의 규제 효과를 가져다주며, 결국 일반화 능력이 향상된다.

## 3.6 다층 퍼셉트론에 의한 인식

$$y = \underset{k}{argmax} o_k (\text{평균적인 복잡도}) \tag{3.26}$$

## 3.7 다층 퍼셉트론의 특성

### 3.7.1 오류 역전파 알고리즘의 빠른 속도

학습 알고리즘의 시간 복잡도: $\Theta((dp + pc)nq)$

### 3.7.2 모든 함수를 정확하게 근사할 수 있는 능력

### 3.7.3 성능 향상을 위한 휴리스틱의 중요성

**실용적인 성능**

# 4 딥러닝 기초

## 4.1 딥러닝의 등장

그레이디언트 소멸gradient vanishing

### 4.1.1 딥러닝의 기술 혁신 요인

### 4.1.2 특징 학습의 부각

수작업 특징hand-crafted feature

특징 학습feature learning

표현 학습representation learning

## 4.2 깊은 다층 퍼셉트론

### 4.2.1 구조와 동작

깊은 MLPDMLP(deep MLP)

$$l\text{번째 층의 연산: } \mathbf{z}^l = \boldsymbol{\tau}_l(\mathbf{U}^l\mathbf{z}^{l-1}), 1 \le l \le L \tag{4.5}$$

### 4.2.2 학습

**오류 역전파 알고리즘** $L$번째 층(출력층)을 위한 gradient 계산식

$$\delta_k^L = \tau_L'(s_k^L)(y_k - o_k), \qquad 1 \le k \le c \tag{4.6}$$

$$\frac{\partial J}{\partial u_{kr}L} = -\delta_k^L z_r^{L-1}, \qquad 0 \le r \le n_{L-1},\ 1 \le k \le c \tag{4.7}$$

$l+1$층의 정보를 이용하여 $l$층의 gradient를 계산하는 공식

$$\delta_j^l = \tau_l'(s_j^l)\sum_{p=1}^{n_{l+1}} \delta_p^{l+1} u_{pj}^{l+1}, \qquad 1 \le j \le n_l \tag{4.8}$$

$$\frac{\partial J}{\partial u_{ji}l} = -\delta_j^l z_i^{l-1}, \qquad 0 \le i \le n_{l-1},\ 1 \le j \le n_l \tag{4.9}$$

**역사적 고찰**

## 4.3 컨볼루션 신경망

### 4.3.1 컨볼루션층

#### 컨볼루션 연산

$$s(i) = z \circledast u = \sum_{x=-(h-1)/2}^{(h-1)/2} z(i+x)u(x) \tag{4.10}$$

$$s(i) = z \circledast u = \sum_{x=-(h-1)/2}^{(h-1)/2} \sum_{x=-(h-1)/2}^{(h-1)/2} z(j+y, i+x)(y, x) \tag{4.11}$$

#### 가중치 공유와 다중 특징 맵 추출

#### 컨볼루션 연산에 따른 CNN의 특성

컨볼루션은 이동에 동변translation equivariant이다. 즉, 신호가 이동하면 이동 정보가 특징 맵에 그대로 반영된다. 수학적으로는 c(t($x$)) = t(c($x$)) (t : 이동 연산, c : 컨볼루션 연산)라 할 수 있다.

#### 큰 보폭에 의한 다운샘플링

#### 텐서에 적용

### 4.3.2 폴링층

풀링의 종류

- 최대 풀링
- 평균 풀링
- 가중치 평균 풀링
- L2 놈 풀링

풀링의 이점

- 요약 통계 추출로 성능 향상
- 속도 향상
- 메모리 효율 증가

풀링의 특성

- 학습으로 알아내야 할 매개변수가 없다.
- 특징 맵마다 독립적으로 풀링 연산을 적용하므로 특징 맵의 개수가 그대로 유지된다.
- 작은 이동에 둔감해지게 한다. 커널의 크기를 키우면 더 둔감해진다.(물체 인식이나 영상 검색 등에 효과적이다.)

[Boureau2010]에서 최대 풀링과 평균 풀링을 비교하고 왜 성능 향상에 기여하는지 이론적으로 분석했다.

### 4.3.3 전체 구조

**빌딩블록**

**LeNet-5**

**가변 크기의 데이터 다루기**

CNN은 DMLP와 달리 가변 크기의 입력을 다룰 수 있다는 장점이 있다.

## 4.4 컨볼루션 신경망 사례연구

### 4.4.1 AlexNet

### 4.4.2 VGGNet

### 4.4.3 GoogLeNet

### 4.4.4 ResNet

## 4.5 생성 모델

### 4.5.1 생성 모델이란?

### 4.5.2 GAN

## 4.6 딥러닝은 왜 강력한가?

**전체 과정을 동시에 초기화**

**깊이의 중요성**

**계층적 특징**

# 5 딥러닝 최적화

## 5.1 목적함수: 교차 엔트로피와 로그우도

### 5.1.1 평균제곱 오차를 다시 생각하기

### 5.1.2 교차 엔트로피 목적함수

$$e = -\sum_{i=1,c}(y_i \log_2 o_i + (1 - y_i \log_2(1 - o_i))) \tag{5.6}$$

### 5.1.3 softmax 활성함수와 로그우도 목적함수

softmax

$$o_j = \frac{e^{s_j}}{\sum_{i=1,c} e^{s_j}} \tag{5.7}$$

log likelihood

$$e = -\log_2 o_y \tag{5.8}$$

## 5.2 성능 향상을 위한 요령

### 5.2.1 데이터 전처리

정규화

$$x_i^{new} = \frac{x_i^{old} - \mu_i}{\sigma_i} \tag{5.9}$$

### 5.2.2 가중치 초기화

### 5.2.3 모멘텀

그레이언트에 스무딩을 가하면 수렴 속도를 개선할 수 있다.

$$\left.\begin{aligned}\mathbf{v} &= \alpha\mathbf{v} - \rho\frac{\partial J}{\partial \Theta} \\ \Theta &= \Theta + \mathbf{v}\end{aligned}\right\} \tag{5.12}$$

네스테로프 모멘텀

$$\begin{aligned}\tilde{\Theta} &= \Theta + \alpha\mathbf{v} \\ \mathbf{v} &= \alpha\mathbf{v} - \rho\frac{\partial J}{\partial \Theta}\bigg|_{\tilde{\Theta}} \\ \Theta &= \Theta + \mathbf{v}\end{aligned} \tag{5.13}$$

### 5.2.4 적응적 학습률

**AdaGrad**

**RMSProp**

가중 이동 평균 기법

$$\mathbf{r} = \alpha\mathbf{r} + (1-\alpha)\mathbf{g} \odot \mathbf{g} \tag{5.14}$$

**Adam**

### 5.2.5 활성함수

$$\begin{aligned}z &= \mathbf{w}^T\tilde{\mathbf{x}} + b \\ y &= \tau(z)\end{aligned} \tag{5.15}$$

tanh는 전 구간에서 미분할 수 있는 장점이 있지만, 값이 커질수록 미분값이 0에 근접하여 학습에 지장을 준다.

$$\mathrm{leakyReLU}(z) = \begin{cases} z, & z \geq 0 \\ az, & z < 0 \end{cases} \tag{5.17}$$

### 5.2.6 배치 정규화

학습 도중에 샘플의 분포가 바뀌는 현상을 covariate shift라 한다.

## 5.3 규제의 필요성과 원리

### 5.3.1 과잉적합에 빠지는 이유와 과잉적합을 피하는 전략

### 5.3.2 규제의 정의

## 5.4 규제 기법

### 5.4.1 가중치 벌칙

규제 항 $R$은 훈련집합과는 무관하고, 가중치의 크기에 제약을 가한다.

$$J_{regularized}(\Theta; \mathbb{X}, \mathbb{Y}) = J(\Theta; \mathbb{X}, \mathbb{Y}) + \lambda R(\Theta) \tag{5.20}$$

**L2 놈**

**선형 회귀에 적용**

선형 회귀

$$J(\mathbf{w}) = \sum_{i=1}^{n} (\mathbf{x_i^T w} - y_i)^2 = \|\mathbf{Xw} - \mathbf{y}\|_2^2 \tag{5.26}$$

리지 회귀

$$\hat{\mathbf{w}} = (\mathbf{X^T X} + 2\lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y} \tag{5.29}$$

**MLP와 DMLP에 적용**

**L1 놈**

### 5.4.2 조기 멈춤

참을성 인자

### 5.4.3 데이터 확대

데이터 확대data augmentation는 잠재적인 변형을 프로그램으로 구현하여 샘플의 수를 강제로 늘리는 기법이다.

이동, 회전, 크기 변환은 선형 변환으로서, 어파인 변환이라고도 한다.

### 5.4.4 드롭아웃

입력층에서 제거될 비율 $P_{input} = 0.2$, 은닉층에서 제거될 비율 $P_{hidden} = 0.5$로 설정하면 적절하다고 보고되어 있다.[Srivastava2014]

### 5.4.5 앙상블 기법

## 5.5 하이퍼 매개변수 최적화

### 5.5.1 격자 탐색과 임의 탐색

임의 탐색이 격자 탐색보다 훨씬 유리하다.

## 5.6 2차 미분을 이용한 최적화

**경사 하강법 다시 보기**

### 5.6.1 뉴턴 방법

### 5.6.2 켤레 그레이디언트 방법

### 5.6.3 유사 뉴턴 방법

**기계 학습에서 2차 미분 정보의 활용**

# 6 비지도 학습

## 6.1 지도 학습과 비지도 학습, 준지도 학습

- 매니폴드 가정: 데이터집합은 하나 또는 여러개의 매니폴드를 구성하며, 모든 샘플은 매니폴드와 가까운 곳에 있다.
- 매끄러움 가정: 샘플은 어떤 요인에 의해 변한다. 이때 매끄러운 곡면을 따라서 변한다.

**비지도 학습에서 사전 지식의 중요성**

## 6.2 비지도 학습

### 6.2.1 비지도 학습의 일반 과업

- 군집화: 유사한 샘플을 모아 같은 그룹으로 묶는 일이다. 몇 개 군집으로 묶을지 알려져 있을수도 있고 아닐수도 있다.
- 밀도 추정: 데이터로부터 확률 분포를 추정하는 일이다. 모수적 추정법과 비모수적 추정법으로 나뉜다.
- 공간 변환: 데이터가 정의된 원래 특징 공간을 저차원 공간 또는 고차 원 공간으로 변환하는 일이다. 새로운 공간은 주어진 목적을 달성하는데 더 유리해야 한다.

### 6.2.2 비지도 학습의 응용 과업

## 6.3 군집화

한 샘플이 하나의 군집에 속하도록 강제하는 방식을 hard clustering이라 하고, 샘플이 군집마다 속하는 정도를 다르게 하는 방식을 soft clustering이라 한다.
군집화를 부류 발견 작업이라 한다.

### 6.3.1 k-평균 알고리즘

**최적화 문제로 해석**

**다중 시작 k-평균 알고리즘**

**EM 기초**

임시로 사용되다 사라지는 변수를 은닉 변수latent variable 라고 한다.

은닉변수의 추정과 매개변수 추정을 번갈아 수행하면서 최적의 해를 찾는 과정을 EM 알고리즘이라고 한다.

### 6.3.2 친밀도 전파 알고리즘

친밀도 전파 알고리즘은 샘플 간의 유사도로부터 책임 행렬 $R$과 가용 행렬 $A$라는 두 종류의 친밀도 행렬을 계산하고, 이 친밀도 정보를 이용하여 군집을 찾는다.

## 6.4 밀도 추정

확률밀도함수 $P(X)$를 구하는 문제를 밀도 추정 문제라 한다.

### 6.4.1 커널 밀도 추정

$$P(x) = \frac{bin(x)}{n} \tag{6.7}$$

식 (6.7)을 이용하여 확률밀도함수를 추정하는 방법을 히스토그램 방법이라고 한다.
이 방법은 단순하여 이해하기 쉽지만 확률밀도함수가 매끄럽지 못하고 계단 모양을 띤다는 심각한 문제점이 있다.

식 (6.8)을 이용하면 이러한 문제점을 해결할 수 있다. 이 식을 사용하는 방법을 커널 밀도 추정법이라고 한다.

$$\left.\begin{aligned} P_h(x) &= \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh^d}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \\ \text{여기서 } K_h(x) &= \frac{1}{h^d} K\left(\frac{x}{h}\right) \end{aligned}\right\} \tag{6.8}$$

### 6.4.2 가우시안 혼합

가우시안을 이용하는 방법은 몇 개의 매개변수로 확률분포를 정의하므로 모수적 방법이지만, 커널 밀도 추정법은 매개변수로 정의되는 일정한 모양의 함수를 사용하지 않으므로 비모수적 방법에 속한다.

### 6.4.3 EM 알고리즘

## 6.5 공간 변환의 이해

## 6.6 선형 인자 모델

### 6.6.1 주성분 분석

주성분 분석PCA(principal component analysis)은 데이터를 원점 중심으로 옮겨 놓는 일부터 시작한다.

$$\left.\begin{array}{ll} x_i - \mu & i = 1, 2, \cdots, n \\ \text{이때 } \mu = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} x_i & \end{array}\right\} \tag{6.19}$$

식 (6.20)은 주성분 분석이 사용하는 변환식이다.

$$\left.z = W^T x \text{이때 } W = (u_1 u_2 \cdots u_q) \text{이고, } u_j = (u_{1j}, u_{2j}, \cdots, u_{dj})^T\right\} \tag{6.20}$$

PCA에 의한 변환은 정보 손실을 일으키는데, 정보 손실이 적을수록 좋은 축이다.

**학습 알고리즘**

PCA의 목적을 정보 손실을 최소화하면서 저차원으로 변환하는 것으로 규정한다. 기계 학습이 할 일은 훈련집합 $\mathbb{X}$가 주어지면 정보 손실을 최소화하는 변환 행렬 $W$를 찾는 것이다. 변환된 훈련집합 $\mathbb{Z}$의 분산이 크면 클수록 정보 손실이 작다고 판단한다.

> 문제 6.1 $\mathbb{Z} = z_1, z_2, \cdots, z_n$의 분산을 최대화하는 $q$개의 축, 즉 $u_1, u_2, \cdots, u_q$를 찾아라. 이 단위 벡터는 식 (6.20)에 따라 변환 행렬 $W$를 구성한다.

### 6.6.2 독립 성분 분석

**블라인드 원음 분리**

**독립성 가정**

**비가우시안 과정**

**ICA 학습**

### 6.6.3 희소 코딩

## 6.7 오토인코더

**동작 원리와 학습**

### 6.7.1 규제 오토인코더

**SAE, DAE, CAE**

소금후추 잡음salt-and-pepper noise

**오토인코더가 알아내는 매니폴드**

### 6.7.2 적층 오토인코더

**층별 예비학습**

층별 예비학습layer-wise pretraining

탐욕 알고리즘greedy algoritim

탐욕 층별 예비학습greedy layer-wise pretraining

**확률 오토인코더**

**층별 예비학습이 딥러닝에 끼친 영향**

## 6.8 매니폴드 학습

### 6.8.1 매니폴드란?

### 6.8.2 IsoMap

### 6.8.3 LLE

### 6.8.4 t-SNE

### 6.8.5 귀납적 학습 모델과 트랜스덕티브 학습 모델

# 7 준지도 학습과 전이 학습

## 7.1 표현 학습의 중요성

### 7.1.1 표현 학습의 대두

### 7.1.2 매니폴드 관찰

### 7.1.3 프라이어를 이용한 변화 인자 풀어내기

## 7.2 내부 표현의 이해

### 7.2.1 컨볼루션 필터의 가시화

### 7.2.2 특징 맵의 가시화

### 7.2.3 영상공간으로 역투영

**최적화를 이용한 역투영**

$$\mathbf{x}_{t+1} = r_{\Theta}(\mathbf{x}_t + \eta \frac{\partial a_i(\mathbf{x})}{\partial \mathbf{x}}) \tag{7.5}$$

**디컨볼루션을 이용한 역투영**

## 7.3 준지도 학습

준지도 학습에서는 레이블이 없는 샘플로부터 정확한 정보를 가급적 많이 끄집어내 성능을 향상시키는 것이 주목적이다.
준지도 학습은 표현 학습과 관련이 깊다.

### 7.3.1 동기와 원리

**레이블이 없는 데이터가 정말 도움이 되는가**

**사람도 준지도 학습을 하는가**

### 7.3.2 알고리즘

**생성 모델**

**현대적 생성 모델**

**자가 학습**

**협동 학습**

**그래프 방법**

**표현 변환**

**밀집지역 회피**

## 7.4 전이 학습

### 7.4.1 과업 전이

**기성 CNN 특징**

**왜 작동할까?**

### 7.4.2 도메인 전이

**도메인 적응**

# 8 순환 신경망

## 8.1 순차 데이터

### 8.1.1 순차 데이터의 표현

**텍스트 순차 데이터의 표현**

### 8.1.2 순차 데이터의 특성

## 8.2 순환 신경망

순차 데이터를 처리하는 신경망은 다음 세 가지 기능을 갖추어야 한다.

- 시간성: 특징을 순서대로 한 번에 하나씩 입력해야 한다.
- 가변 길이: 길이가 T인 샘플을 처리하려면 은닉층이 T번 나타나야 한다. T는 가변적이다.
- 문맥 의존성: 이전 특징 내용을 기억하고 있다가 적절한 순간에 활용해야 한다.

### 8.2.1 구조

RNN은 순환 에지를 가짐으로서 시간성, 가변 길이, 문맥 의존성의 세 가지 기능을 갖춘다. 순환 에지는 t-1 순간에 발생한 정보를 t 순간으로 전달한다.

### 8.2.2 동작

### 8.2.3 BPTT 학습

**RNN과 MLP의 유사성과 차별성**

**목적함수 정의**

$$J(\Theta) = \sum_{t=1}^{T} J^{(t)}(\Theta) \tag{8.10}$$

$$\text{평균제곱 오차: } J^{(t)}(\Theta) = \sum_{j=1}^{q} (y_j^{(t)} - y_j'^{(t)})^2 \tag{8.11}$$

$$\text{교차 엔트로피: } J^{(t)}(\Theta) = -\mathbf{y}^{(t)} \log \mathbf{y}'^{(t)} = -\sum_{j=1}^{q} y_j^{(t)} \log y_j'^{(y)} \tag{8.12}$$

$$\text{로그 우도: } J^{(t)}(\Theta) = -\log y'^{(t)} \tag{8.13}$$

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}} J(\Theta) = \underset{\Theta}{\operatorname{argmin}} \sum_{t=1}^{T} J^{(t)}(\Theta) \tag{8.14}$$

**그레이디언트 계산**

**BPTT 알고리즘**

### 8.2.4 양방향 RNN

## 8.3 장기 문맥 의존성

관련된 요소가 멀리 떨어져 있는 경우를 장기 의존성이라고 한다.
RNN의 gradient 소멸 현상에 관한 논문[Pascanu2013]

## 8.4 LSTM

LSTM[Long Short Term Memory]은 여러 종류의 게이트가 있어 입력을 선별적으로 허용하고 계산 결과를 선별적으로 출력할 수 있다[Hochreiter1997]. GRU[gated recurrent unit]는 성능 저하를 최소로 유지하면서 LSTM을 단순화한 모델이다[Chung2014].

### 8.4.1 게이트를 이용한 영향력 범위 확장

### 8.4.2 LSTM의 동작

**행렬 표기**

### 8.4.3 망각 게이트와 핍홀

핍홀은 메모리 블록의 내부 상태를 3개의 게이트에 알려 주는 역할을 한다.

## 8.5 응용 사례

순환 신경망은 주로 가변 길이의 패턴을 처리하는 데 활용된다.

### 8.5.1 언어 모델

**n-그램**

**전방 신경망 모델**

**순환 신경망 모델**

**생성 모델로 활용**

### 8.5.2 기계 번역

### 8.5.3 영상 주석 생성

영상을 해석하고 결과를 문장으로 변환해 출력하는 응용 과업을 영상 주석 생성이라고 한다. CNN은 영상을 분석하고 인식하는데 적합하고 RNN은 언어 모델을 구축하여 문장을 생성하는 데 적합하므로 대부분 둘을 결합하여 사용한다.
ImageNet으로 지도학습한 CNN에서 오른쪽 분류층을 떼고 마지막 컨볼루션층에 완전연결층을 붙이고 완전연결층의 출력은 RNN에 연결한다.

# 9 강화 학습

## 9.1 강화 학습의 원리와 성질

### 9.1.1 계산 모형

#### 상태, 행동, 보상

$$\text{강화 학습의 핵심 연산 } f : (s_t, a_t) \rightarrow (s_{t+1}, r_{t+1}) \quad (9.1)$$
$$\text{where } s = \text{상태}, a = \text{행동}, r = \text{보상}$$

#### 에이전트와 환경

MDP는 문제를 정의할 때 따라 주어지는 정보이다.

#### 정책

에이전트는 일정한 확률 규칙으로 행동을 결정해야 하는데, 이 규칙을 정책이라 한다.

### 9.1.2 탐험과 탐사

최적화 문제를 풀 때 탐험과 탐사 갈등이 발생한다.
탐험은 전체 공간을 골고루 찾아보는 전략이고, 탐사는 특정한 곳 주위를 집중적으로 찾아보는 전략이다. 탐험은 시간이 너무 오래 걸리고, 탐험은 더 좋은 해를 놓치고 지역 최적점에 머무는 문제가 있다.

### 9.1.3 마르코프 결정 프로세스

강화 학습에서 가장 중요한 연산은 어떤 상태에서 가장 유리한 행동을 선택하는 것이다.

#### 마르코프 성질

스토캐스틱 프로세스에서 미래 상태의 조건부 확률 분포가 선행되었던 이벤트의 시퀀스와는 상관없이, 오직 현재의 상태에 의존하면, 해당 프로세스는 마르코프 성질Markov property을 만족한다고 할 수 있다.
마르코프 성질을 수학 공식으로 표현하면 다음과 같다.

$$P(s_{t+1}, r_{t+1} \mid s_0, a_0, r_1, s_1, a_1, \cdots, r_{t-1}, s_{t-1}, a_{t-1}, r_t, s_t, a_t) = P(s_{t+1}, r_{t+1} \mid s_t, a_t) \tag{9.2}$$

표기를 간단히 하면 다음과 같다.

$$P(s_{t+1}, r_{t+1} \mid s_t, a_t) = P(s', r \mid s, a) \tag{9.3}$$

#### 환경 모델로서 MDP

$$\text{MDP 확률분포: } P(s', r \mid s, a), \forall \in \mathcal{S}, \forall a \in \mathcal{A}, \forall s' \in \mathcal{S}, \forall \in \mathcal{R} \tag{9.4}$$

식 (9.4)를 MDP의 확률분포라고 한다. 이 확률분포는 Markov property를 만족한다.

## 9.2 정책과 가치함수

강화 학습의 핵심은 좋은 정책을 찾아내는 것이다.
정책 공간이 너무 방대하여 최적 정책을 직접 찾는 접근방법은 무모하다.

### 9.2.1 정책

정책이란 상태 $s$에서 행동 $a$를 취할 확률을 모든 상태와 행동에 대해 명시한 것을 말한다.

$$\pi(a|s) = P(a|s), \forall s, \forall a \tag{9.5}$$

#### 최적 정책

#### 최적 정책 찾기

$$\hat{\pi} = \underset{\pi}{\operatorname{argmax}}\, goodness(\pi) \tag{9.7}$$

서로 다른 정책의 집합을 정책 공간이라 하고, 상태 공간보다 훨씬 방대하다.
exhaustive search는 적용할 수 없다.
정책 공간을 직접 탐색하는 대신 최적 가치함수를 탐색한다.

### 9.2.2 가치함수

가치함수는 특정 정책의 goodness를 평가하는 함수이다.
특정 정책에서 모든 상태의 goodness를 평가한다.
goodness는 상태 $s$로부터 종료 상태에 이르기까지 누적 보상의 추정치이다.
식 9.7을 좀 더 구체적으로 쓰면 식 (9.8)이 된다.

$$\hat{\pi} = \underset{\pi}{\operatorname{argmax}}\, v_\pi(s), \forall_S \in \mathcal{S} \tag{9.8}$$

가치함수를 일반적인 수식으로 표현하면 식(9.9)이다.

$$v_\pi(s) = \sum_{\text{s에서 출발하는 모든 경로 z}} P(z)\mathbb{r}(z) \tag{9.9}$$

**에피소드 과업과 영구 과업**
유한 경로(episode task): $z : (s_t, r_t) \xrightarrow{a_t} (s_{t+1}, r_{t+1}) \xrightarrow{a_{t+1}} (s_{t+2}, r_{t+2}) \xrightarrow{a_{t+2}} \cdots \xrightarrow{a_{T-1}} (s_T, r_T)$
무한 경로(continuing task): $z : (s_t, r_t) \xrightarrow{a_t} (s_{t+1}, r_{t+1}) \xrightarrow{a_{t+1}} (s_{t+2}, r_{t+2}) \xrightarrow{a_{t+2}} (s_{t+3}, r_{t+3}) \cdots$

$$\mathbb{r}(z) = r_{t+1} + \gamma r_{t+1} + \gamma_2 r_{t+2} \cdots = \sum_{k=1}^{\infty} \gamma^{k-1} r_{t+k} \tag{9.11}$$

**가치함수 추정을 위한 순환식**

$$v_\pi(s) = \sum_{a \in (A)(s)} P(a|s)(r + v_\pi(s')), \forall_S \in (S) \tag{9.12}$$

**스토캐스틱 프로세스에서 가치함수 추정**
스토캐스틱 프로세스에서는 식 (9.12)보다 복잡한 수식을 사용해야 한다.

$$\sum_{a \in \mathcal{A}} P(a|s) \sum_{s'} \sum_{r} P(s', r|s, a)(r + v_\pi(s')), \forall_S \in \mathcal{S} \tag{9.12}$$

무한 경로를 가진 문제라면 할인율을 적용한 식 (9.14)를 사용한다.

$$v_\pi(s) = \sum_{a \in \mathcal{A}(s)} P(a|s) \sum_{s'} \sum_{r} P(s', r|s, a)(r + \gamma v_\pi(s')), \forall_S \in \mathcal{S} \tag{9.14}$$

식 (9.13)과 식 (9.14)는 상태 가치함수라 하고, 식 (9.15)는 상태-행동 가치함수라 한다.

$$q_\pi(s, a) = \sum_{s'} \sum_{r} P(s', r|s, a)(r + \gamma v_\pi(s')), \forall_S \in \mathcal{S}, \forall_a \in \mathcal{A} \tag{9.15}$$

### 9.2.3 최적 가치함수

식 (9.16)은 최적 가치함수를 정의한다.
식 (9.17)은 최적 가치함수를 추정하는 방법이다.

$$\hat{v}(s) = \max_{a \in \mathcal{A}(s)} \sum_{s'} \sum_{r} P(s', r|s, a)(r + \hat{v}(s')), \forall_S \in \mathcal{S} \tag{9.17}$$

식 (9.18)은 무한 경로를 위해 할인율을 적용한 수식이다.

$$\hat{v}(s) = \max_{a \in \mathcal{A}(s)} \sum_{s'} \sum_{r} P(s', r|s, a)(r + \gamma\hat{v}(s')), \forall_S \in \mathcal{S} \tag{9.18}$$

## 9.3 동적 프로그래밍

**스토캐스틱 동적 프로그래밍**

### 9.3.1 정책 반복 알고리즘

### 9.3.2 가치 반복 알고리즘

다른 상태의 값을 보고 자신의 값을 갱신하는 방식을 부트스트랩 이라고 한다.
부트스트랩 방식에서는 모든 상태가 이전 값을 이용하여 현재 값을 계산해야 공평하지만, 제자리 연산으로 구현하면 수렴 속도가 빨라지는 장점이 있다.
동적 프로그래밍은 상태와 행동의 개수가 적은 경우에만 사용한다.

## 9.4 몬테카를로 방법

### 9.4.1 훈련집합의 수집과 정책 평가

$$e = [s_0, r_0]a_0[s_1, r_1]a_1[s_2, r_2]a_2 \cdots [s_t, r_t]a_t \cdots [s_T, r_T] \tag{9.20}$$

**정책 평가**

$$v_\pi(s) = \frac{1}{|Z(s)|} \sum_{z \in Z(s)} \mathbb{r}(z),\ \forall_S \in \mathcal{S} \tag{9.21}$$

$$q_\pi(s, a) = \frac{1}{|Z(s, a)|} \sum_{z \in Z(s,a)} \mathbb{r}(z),\ \forall_S \in \mathcal{S},\ \forall_a \in \mathcal{A} \tag{9.22}$$

### 9.4.2 최적 정책 탐색

**탐험과 탐사 조절**

**탐험형 시작**

에피소드를 생성할 때 모든 상태-행동 쌍이 골고루 발생하도록 하는 방법을 탐험형 시작이라 한다. 주류에서 벗어난 상태-행동에 일정한 확률을 배정하여 선택될 가능성을 열어두는 것을 $\varepsilon$-소프트라고 한다.

**$\varepsilon$-소프트**

**몬테카를로 방법의 특성**

- 환경 모델이 없어도 된다.
- 부트스트랩 방식이 아니므로 관심 있는 상태로만 구성된 부분집합에 국한하여 최적 가치와 최적 정책을 추정할 수 있다.
- 마르코프 성질에서 크게 벗어나는 상황에서도 성능 저하가 비교적 적다.
- 동적 프로그래밍과 시간차 학습은 부트스트랩을 사용하지만, 몬테카를로 방법은 부트스트랩을 사용하지 않는다.

## 9.5 시간차 학습

### 9.5.1 정책 평가

식 (9.21)에 $s_t$를 대입하면 다음과 같다.

$$v_\pi(s_t) = \frac{1}{Z(s_t)} \sum_{z \in Z(s_t)} \mathbb{r}(z)$$

샘플 $z_t$가 $k$번째로 추가되었다면, 이 식을 다음과 같이 쓸 수 있다.

$$v_\pi(s_t) = v_\pi(s_t) + \rho(\mathbb{r}_{new} - v_{pi}(s_t)) \tag{9.23}$$

식 (9.24)는 시간차 학습TD(temporal difference) learning이 사용하는 식이다.

$$v_\pi(s_t) = v_[\pi](s_t) + \rho((r_{t+1} + \gamma v_\pi(s_{t+1})) - v_\pi(s_t)) \tag{9.24}$$

상태-행동 가치함수를 추정하려면 식 (9.25)를 사용한다.

$$q_\pi(s_t, a_t) = q_\pi(s_t, a_t) + \rho((r_{t+1} + \gamma q_\pi(s_{t+1}, a_{t+1})) - q_\pi(s_t, a_t)) \tag{9.25}$$

**정책 평가**

### 9.5.2 Sarsa

### 9.5.3 Q-학습

$$q_\pi(s_t, a_t) + \rho((r_{t+1} + \max_a q_\pi(s_{t+1}, a)) - q_\pi(s_t, a_t)) \tag{9.26}$$

## 9.6 근사 방법

lookup table
현재 강화 학습은 근사 방법보다 신경망을 주로 사용한다.

## 9.7 응용 사례

### 9.7.1 TD-gammon

**강화 학습과 보드게임**

### 9.7.2 DQN: 아타리 비디오 게임

# 10 확률 그래피컬 모델

## 10.1 확률과 그래프의 만남

### 10.1.1 그래프 표현

### 10.1.2 그래프 분해와 확률 표현

## 10.2 베이지안 네트워크

**세 가지 주요 문제**

### 10.2.1 간단한 예제

### 10.2.2 그래프 분해

### 10.2.3 d-분리

**조건부 독립**

**체인의 폐쇄**

**d-분리**
베이지안 네트워크는 선형, 분기, 합류의 세 가지 연결 패턴으로 구성된다. 이들의 특성과 동작을 잘 분석하면 복잡한 그래프에서의 조건부 독립을 알아낼 수 있다.

### 10.2.4 확률 추론

**d-분리와 확률 추론**

**정확한 해 구하기**

**근사 추론**

## 10.3 마르코프 랜덤필드

이름에 마르코프가 붙은 이유는 이웃한 노드 사이에만 직접적인 상호작용을 허용하기 때문이다. 확률변수들이 동일한 자격으로 영향을 주고받으면서 필드를 형성하므로 랜덤필드이다.

### 10.3.1 동작 원리

모든 노드 쌍이 에지를 가지는 완전 부분그래프를 클릭clique이라고 한다. 노드를 추가하면 완전 그래프를 유지하지 못하는 클릭을 극대 클릭maximal clique이라고 한다.

같은 종류의 노드 사이에는 에지를 허용하지 않는 구조를 제한 볼츠만 기계RBM 라고 부른다. DBNdeep belief network

### 10.3.2 사례 연구: 영상 잡음 제거

마르코프 랜덤필드는 주로 영상처리와 컴퓨터비전에서 잡음 제거, 영상 복원, 에지 검출, 텍스쳐 분석, 스테레오 비전, 영상 분할 등의 문제를 풀 때 활용한다.[Li2009, Blake2011].

#### 에너지함수 공식화

#### 최적화 알고리즘

## 10.4 RBM과 DBN

### 10.4.1 RBM의 구조와 원리

#### RBM 구조

#### 에너지와 확률분포

### 10.4.2 RBM 학습

#### 목적함수

#### 대조 발산 알고리즘

#### RBM의 응용

### 10.4.3 DBN

#### RBM 쌓기

#### DBN의 응용

# 11 커널 기법

## 11.1 커널 트릭

### 커널함수와 커널 트릭

정의 11 - 1 **커널함수**
원래 특징 공간 $\mathcal{L}$에 정의된 두 특징 벡터 $x$와 $z$에 대해 $K(x,z) = \Phi(x) \cdot \Phi(z)$ 인 변환함수 $\Phi$가 존재하면 $K(x,z)$를 커널함수라 부른다.

커널 트릭이란 $\Phi(x)$로 변환한 $\mathcal{H}$공간에서 내적 연산을 원래 특징 공간 $\mathcal{L}$에서 커널함수 계산으로 대치하는 것이다. 다시 말해 특징 벡터를 명시적으로 $\mathcal{H}$로 매핑하지 않고 커널함수를 이용하여 $\mathcal{H}$공간의 내적을 얻는다.

### 널리 쓰이는 세 가지 커널함수

$$\text{다항식 커널: } K(x,z) = (x \cdot z + 1)^p \tag{11.6}$$

$$\text{RBF 커널: } K(x,z) = exp\left(\frac{-\|x-z\|_2^2}{2\sigma^2}\right) \tag{11.7}$$

$$\text{하이퍼볼릭 탄젠트 커널: } K(x,z) = \tanh(ax \cdot z + \beta) \tag{11.8}$$

어떤 함수 K가 커널함수인지를 확인할 때에는 Mercer 정리를 사용한다[Burges1998].

## 11.2 커널 리지 회귀

### 쌍대 문제로 변환

### 메모리 기반 예측

커널 트릭은 메모리 기반 접근방법이다.

## 11.3 커널 PCA

## 11.4 SVM 분류

### 여백을 이용한 일반화 능력 향상

### 11.4.1 선형 SVM

이진 선형 분류기의 결정 초평면은 식 (11.23)으로 표현할 수 있다.

$$d(\mathbf{x}) = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b = \mathbf{w}^T \mathbf{x} + b = 0 \tag{11.23}$$

**결정 초평면의 수학적 특성**
식 (11.23)은 SVM 수식을 유도하는 데 도움이 되는 몇 가지 수학적 특성이 있다.

- d(x) = 0은 특징 공간을 부분 공간 2개로 분할한다.
- 식 (11.23)에 상수 $c$를 곱한 $c\mathbf{w}^T\mathbf{x} + cb = 0$도 동일한 초평면이다.
- $\mathbf{w}$는 초평면에 수직인 법선벡터이고 바이어스 $b$는 초평면의 위치를 지정한다.
- 점 $\mathbf{x}$에서 초평면까지 거리는 식 (11.24)로 구한다.

$$h = \frac{|d(\mathbf{x})|}{\|\mathbf{w}\|_2} \tag{11.24}$$

**선형 분리 가능한 상황**

**소프트 여백: 선형 분리가 불가능한 상황**
분할 띠 안에 샘플을 허용하는 아이디어를 소프트 여백soft margin 이라고 한다.

### 11.4.2 비선형 SVM

**학습 알고리즘 구현**

**예측 알고리즘**

**오픈 소스와 활용 가이드라인**

### 11.4.3 c-부류 SVM

## 11.5 SVM 회귀

# 12 앙상블 방법

## 12.1 동기와 원리

### 12.1.1 앙상블을 사용하는 이유

- 나쁜 운을 피할 수 있다.
- 성능 향상을 꾀할 수 있다.
- 데이터 양/질에 따른 어려움을 극복할 수 있다.
- 다중 센서 시스템에 효과적이다.
- 점진 학습이 가능하다.

### 12.1.2 요소 분류기의 다양성

## 12.2 재샘플링 기법

### 12.2.1 배깅

배깅은 부트스트랩 아이디어를 앙상블 생성에 적용한 알고리즘이다.

### 12.2.2 부스팅

## 12.3 결정 트리와 랜덤 포리스트

### 12.3.1 결정 트리

#### 노드에서의 질문

결정 트리는 자식이 최대 2개인 이진 트리를 사용한다.

#### 학습 알고리즘

#### 예측 알고리즘

### 12.3.2 랜덤 포리스트

결정 트리는 신경망이나 SVM등과 같은 다른 모델에 비해 정확도가 낮다. 따라서 단독으로 쓰기보다 앙상블의 요소 분류기로 많이 쓰인다.
랜덤 포리스트는 결정 트리를 배깅으로 결합한 것인데, 가장 널리 활용되는 결정 트리의 앙상블 모델이다. 튜토리얼 논문[Breiman2001, Criminisi2011]

## 12.4 앙상블 결합

### 12.4.1 부류 레이블

### 12.4.2 부류 순위

### 12.4.3 부류 확률

## 12.5 딥러닝과 앙상블

### 12.5.1 평균 기법을 이용한 앙상블

### 12.5.2 암시적 앙상블

### 12.5.3 깊은 랜덤 포리스트

결정 트리의 학습 알고리즘은 greedy algorithm이므로 지역 최적점을 찾는다. 이러한 한계를 극복하는 전역 최적화 기법이 몇 가지 개발되어 있다.

랜덤 포리스트와 CNN을 상호 변환하는 알고리즘이 있다[Richmond2015].