기계 학습 5.4 규제 기법

2021210088 허지혜

5.4.1 가중치 벌칙

- 명시적 규제와 암시적 규제

명시적 규제 : 가중치 감쇠나 드롭아웃처럼 목적함수나 신경망 구조를 직접 수정하는 방식
암시적 규제 : 조기멈춤, 데이터 증대, 잡음 추가, 앙상블처럼 간접적으로 영향을 미치는 방식

(식 5.19)를 관련 변수가 드러나도록 다시 쓰면 다음과 같다.

$$\underbrace{J_{regularized}(\boldsymbol{\Theta}; \mathbb{X}, \mathbb{Y})}_{\text{규제를 적용한 목적함수}} = \underbrace{J(\boldsymbol{\Theta}; \mathbb{X}, \mathbb{Y})}_{\text{목적함수}} + \lambda \underbrace{R(\boldsymbol{\Theta})}_{\text{규제 항}} \qquad (5.20)$$

훈련 집합 $\mathbb{X}, \mathbb{Y}$ 에 영향을 받음

큰 가중치에 벌칙을 가해서
작은 가중치를 유지하려고 주로 L2, L1놈을 사용함

규제항은 훈련집합과 무관하며, 데이터 생성 과정에 내재한 사전 지식에 해당.
규제항은 매개변수를 작은 값으로 유지하므로 모델의 용량을 제한하는 역할을 한다.

5.4.1 가중치 벌칙

- L2 놈

규제항 R을 L2 놈을 사용하는 규제 기법을 ' 가중치 감쇠' 라고 부른다.

$$\underbrace{J_{regularized}(\boldsymbol{\Theta};\mathbb{X},\mathbb{Y})}_{\text{규제를 적용한 목적함수}} = \underbrace{J(\boldsymbol{\Theta};\mathbb{X},\mathbb{Y})}_{\text{목적함수}} + \lambda \underbrace{\|\boldsymbol{\Theta}\|_2^2}_{\text{규제 항}} \quad (5.21)$$

선형 회귀시 리지 회귀라고 한다.

(식 5.21)를 그레디언트 계산을 하면 다음과 같다.

$$\nabla J_{regularized}(\boldsymbol{\Theta};\mathbb{X},\mathbb{Y}) = \nabla J(\boldsymbol{\Theta};\mathbb{X},\mathbb{Y}) + 2\lambda\boldsymbol{\Theta} \quad (5.22)$$

(식 5.22)를 이용하여 매개변수를 갱신하는 수식은 다음과 같다.

$$\begin{aligned}\boldsymbol{\Theta} &= \boldsymbol{\Theta} - \rho\nabla J_{regularized}(\boldsymbol{\Theta};\mathbb{X},\mathbb{Y}) \\ &= \boldsymbol{\Theta} - \rho(\nabla J(\boldsymbol{\Theta};\mathbb{X},\mathbb{Y}) + 2\lambda\boldsymbol{\Theta}) \\ &= (1-2\rho\lambda)\boldsymbol{\Theta} - \rho\nabla J(\boldsymbol{\Theta};\mathbb{X},\mathbb{Y})\end{aligned} \longrightarrow \boldsymbol{\Theta} = (1-2\rho\lambda)\boldsymbol{\Theta} - \rho\nabla J \quad (5.23)$$

5.4.1 가중치 벌칙

$$\begin{aligned}\boldsymbol{\Theta} &= \boldsymbol{\Theta} - \rho\nabla J_{regularized}(\boldsymbol{\Theta}; \mathbb{X}, \mathbb{Y}) \\ &= \boldsymbol{\Theta} - \rho(\nabla J(\boldsymbol{\Theta}; \mathbb{X}, \mathbb{Y}) + 2\lambda\boldsymbol{\Theta}) \\ &= (1 - 2\rho\lambda)\boldsymbol{\Theta} - \rho\nabla J(\boldsymbol{\Theta}; \mathbb{X}, \mathbb{Y})\end{aligned} \longrightarrow \boldsymbol{\Theta} = (1 - 2\rho\lambda)\boldsymbol{\Theta} - \rho\nabla J \quad (5.23)$$

$\lambda = 0$ 를 두면 규제를 적용하지 않은 원래 식 $\Theta = \Theta - \rho\nabla J$ 가 된다.
가중치 감쇠는 단지 Θ 에 $(1 - 2\rho\lambda)$ 를 곱해주는 셈이다.
예를 들어 $\rho = 0.01, \lambda = 2.0$ $(1 - 2\rho\lambda) = 0.96$ 이다.

최종해를 원점 가까이 당기는 효과를 준다.(가중치를 작게 유지한다.)

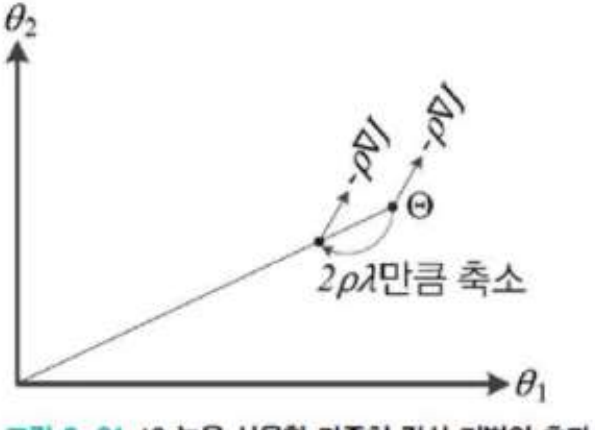


그림 5-21 *L2* 놈을 사용한 가중치 감쇠 기법의 효과

5.4.1 가중치 벌칙

- 선형 회귀에 적용

선형 회귀는 훈련집합 $\mathbb{X} = \{X_1, X_2, \ldots X_n\}, \mathbb{Y} = \{y_1, y_2, \ldots y_n\}$ 이 주어지면 (식 5.24)를 풀어서 $w = (w_1, w_2, \ldots, w_d)^T$ 를 구하는 문제이다. 이때의 $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})^T$ 이다.

$$w_1 x_{i1} + w_2 x_{i2} \cdots + w_d x_{id} = \mathbf{x}_i^{\mathrm{T}} \mathbf{w} = y_i, \qquad i = 1,2,\cdots,n \tag{5.24}$$

(식 5.24)를 행렬식으로 바꿔 쓰면,

$$\mathbf{Xw} = \mathbf{y} \tag{5.25}$$

가중치 감쇠를 적용한 목적함수는 다음과 같다.

$$J_{regularized}(\mathbf{w}) = \|\mathbf{Xw} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|_2^2 = (\mathbf{Xw} - \mathbf{y})^{\mathrm{T}}(\mathbf{Xw} - \mathbf{y}) + \lambda \|\mathbf{w}\|_2^2 \tag{5.27}$$

5.4.1 가중치 벌칙

(식 5.27)을 미분하여 0으로 놓으면,

$$\frac{\partial J_{regularized}}{\partial \mathbf{w}} = \mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w} - \mathbf{X}^{\mathrm{T}}\mathbf{y} + 2\lambda\mathbf{w} = \mathbf{0} \implies (\mathbf{X}^{\mathrm{T}}\mathbf{X} + 2\lambda\mathbf{I})\mathbf{w} = \mathbf{X}^{\mathrm{T}}\mathbf{y} \qquad (5.28)$$

(식 5.28)을 정리하면,

$$\hat{\mathbf{w}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + 2\lambda\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$$ 훈련 집합으로부터 구한 최적값 (5.29)

공분산 행렬 X^TX 의 대각 요소가 2λ 만큼씩 증가 -> 역행렬을 곱하므로 가중치를 축소하여 원점으로 당기는 효과를 준다.

새로운 특징벡터가 입력되면 (식 5.30)을 이용하여 예측할 수 있다.

$$y = \mathbf{x}^{\mathrm{T}}\hat{\mathbf{w}} \qquad (5.30)$$

5.4.1 가중치 벌칙

예제 5-1 **리지 회귀**

훈련집합 $\mathbb{X} = \{\mathbf{x}_1 = \begin{pmatrix}1\\1\end{pmatrix}, \mathbf{x}_2 = \begin{pmatrix}2\\3\end{pmatrix}, \mathbf{x}_3 = \begin{pmatrix}3\\3\end{pmatrix}\}$, $\mathbb{Y} = \{y_1 = 3.0, y_2 = 7.0, y_3 = 8.8\}$이 주어졌다고 가정하자. 특징 벡터가 2차원이므로 d=2이고 샘플이 3개이므로 n=3이다. 훈련집합으로 설계행렬 **X**와 레이블 행렬 **y**를 다음과 같이 쓸 수 있다.

$$\mathbf{X} = \begin{pmatrix}1 & 1\\2 & 3\\3 & 3\end{pmatrix}, \qquad \mathbf{y} = \begin{pmatrix}3.0\\7.0\\8.8\end{pmatrix}$$

이 값들을 식 (5.29)에 대입하여 다음과 같이 $\hat{\mathbf{w}}$을 구할 수 있다. 이때 $\lambda = 0.25$라 가정하자.

$$\hat{\mathbf{w}} = \left(\begin{pmatrix}1 & 2 & 3\\1 & 3 & 3\end{pmatrix}\begin{pmatrix}1 & 1\\2 & 3\\3 & 3\end{pmatrix} + \begin{pmatrix}0.5 & 0\\0 & 0.5\end{pmatrix}\right)^{-1}\begin{pmatrix}1 & 2 & 3\\1 & 3 & 3\end{pmatrix}\begin{pmatrix}3.0\\7.0\\8.8\end{pmatrix} = \begin{pmatrix}1.4916\\1.3607\end{pmatrix}$$

따라서 하이퍼 평면은 $y = 1.4916x_1 + 1.3607x_2$이다. 새로운 샘플로 $\mathbf{x} = (5 \quad 4)^T$가 입력되면 식 (5.30)을 이용하여 12.9009를 예측한다.

5.4.1 가중치 벌칙

- MLP와 DMLP에 적용

$$\left.\begin{aligned}\mathbf{U}^1 &= \mathbf{U}^1 - \rho\frac{\partial J}{\partial \mathbf{U}^1}\\ \mathbf{U}^2 &= \mathbf{U}^2 - \rho\frac{\partial J}{\partial \mathbf{U}^2}\end{aligned}\right\} \quad (3.21) \longrightarrow \left.\begin{aligned}\mathbf{U}^1 &= (1-2\rho\lambda)\mathbf{U}^1 - \rho\frac{\partial J}{\partial \mathbf{U}^1}\\ \mathbf{U}^2 &= (1-2\rho\lambda)\mathbf{U}^2 - \rho\frac{\partial J}{\partial \mathbf{U}^2}\end{aligned}\right\} \quad (5.31)$$

[알고리즘 3-4]에 적용하면,

13. for (k=1 to c) for (j=0 to ρ) $u_{kj}^2 = u_{kj}^2 - \rho\Delta u_{kj}^2$ // 가중치 감쇠 적용하지 않은 원래 알고리즘
14. for (j=1 to ρ) for (i=0 to d) $u_{ji}^1 = u_{ji}^1 - \rho\Delta u_{ji}^1$

⇓

13. for (k=1 to c) for (j=0 to ρ) $u_{kj}^2 = (1-2\rho\lambda)u_{kj}^2 - \rho\Delta u_{kj}^2$ // 가중치 감쇠 적용한 알고리즘
14. for (j=1 to ρ) for (i=0 to d) $u_{ji}^1 = (1-2\rho\lambda)u_{ji}^1 - \rho\Delta u_{ji}^1$

5.4.1 가중치 벌칙

[알고리즘 3-6]에 적용하면,

14. $\mathbf{U}^2 = \mathbf{U}^2 - \rho\frac{\Delta \mathbf{U}^2}{t}$ // 가중치 감쇠 적용하지 않은 원래 알고리즘

15. $\mathbf{U}^1 = \mathbf{U}^1 - \rho\frac{\Delta \mathbf{U}^1}{t}$

⇓

14. $\mathbf{U}^2 = (1 - 2\rho\lambda)\mathbf{U}^2 - \rho\frac{\Delta \mathbf{U}^2}{t}$ // 가중치 감쇠 적용한 알고리즘

15. $\mathbf{U}^1 = (1 - 2\rho\lambda)\mathbf{U}^1 - \rho\frac{\Delta \mathbf{U}^1}{t}$

DMLP를 위한 [알고리즘 4-1]에 적용하면,

16. for (l=L to 1) // 가중치 감쇠 적용하지 않은 원래 알고리즘

17. for (j=1 to n_l) for (i=0 to n_{l-1}) $u_{ji}^l = u_{ji}^l - \rho\left(\frac{1}{t}\right)\Delta u_{ji}^l$

⇓

16. for (l=L to 1) // 가중치 감쇠 적용한 알고리즘

17. for (j=1 to n_l) for (i=0 to n_{l-1}) $u_{ji}^l = (1 - 2\rho\lambda)u_{ji}^l - \rho\left(\frac{1}{t}\right)\Delta u_{ji}^l$

5.4.1 가중치 벌칙

- L1 놈

▪ 규제 항으로 $L1$ 놈을 적용하면, ($L1$ 놈은 $\|\boldsymbol{\Theta}\|_1 = |\theta_1| + |\theta_2| + \cdots$)

$$\underbrace{J_{regularized}(\boldsymbol{\Theta}; \mathbb{X}, \mathbb{Y})}_{\text{규제를 적용한 목적함수}} = \underbrace{J(\boldsymbol{\Theta}; \mathbb{X}, \mathbb{Y})}_{\text{목적함수}} + \lambda \underbrace{\|\boldsymbol{\Theta}\|_1}_{\text{규제 항}} \tag{5.32}$$

▪ 식 (5.32)를 미분하면,

$$\nabla J_{regularized}(\boldsymbol{\Theta}; \mathbb{X}, \mathbb{Y}) = \nabla J(\boldsymbol{\Theta}; \mathbb{X}, \mathbb{Y}) + \lambda \mathbf{sign}(\boldsymbol{\Theta}) \tag{5.33}$$

벡터의 요소가 양수이면 1
음수이면 -1을 가지는 벡터

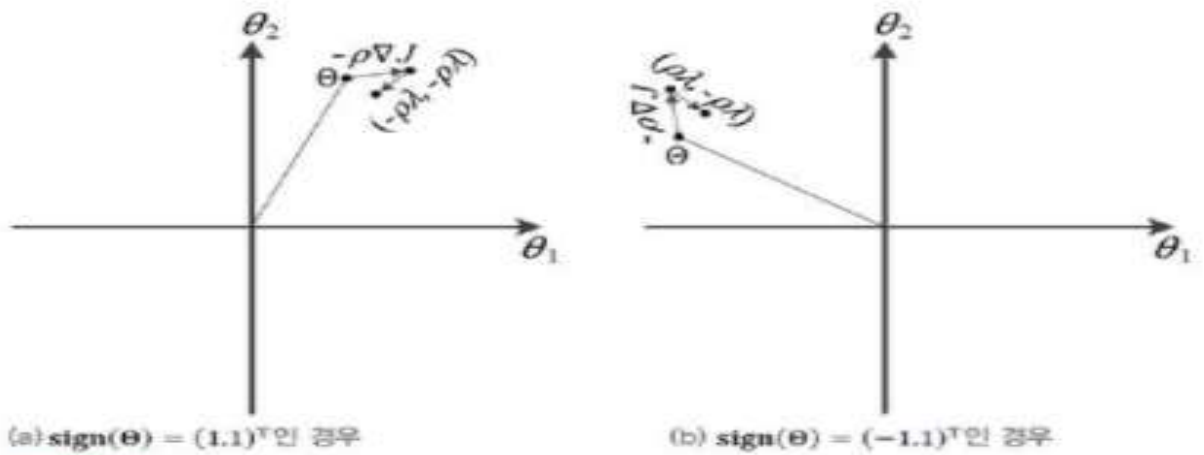
매개변수를 갱신하는 식에 대입하면 다음과 같다.

$$\begin{aligned}\boldsymbol{\Theta} &= \boldsymbol{\Theta} - \rho \nabla J_{regularized}(\boldsymbol{\Theta}; \mathbb{X}, \mathbb{Y}) \\ &= \boldsymbol{\Theta} - \rho(\nabla J(\boldsymbol{\Theta}; \mathbb{X}, \mathbb{Y}) + \lambda \mathbf{sign}(\boldsymbol{\Theta})) \\ &= \boldsymbol{\Theta} - \rho \nabla J(\boldsymbol{\Theta}; \mathbb{X}, \mathbb{Y}) - \rho\lambda \mathbf{sign}(\boldsymbol{\Theta})\end{aligned}$$

5.4.1 가중치 벌칙

- 매개변수를 갱신하는 식

$$\boldsymbol{\Theta} = \boldsymbol{\Theta} - \rho\nabla J - \rho\lambda \mathbf{sign}(\boldsymbol{\Theta}) \qquad (5.34)$$



(a) $\mathbf{sign}(\boldsymbol{\Theta}) = (1.1)^T$인 경우 (b) $\mathbf{sign}(\boldsymbol{\Theta}) = (-1.1)^T$인 경우

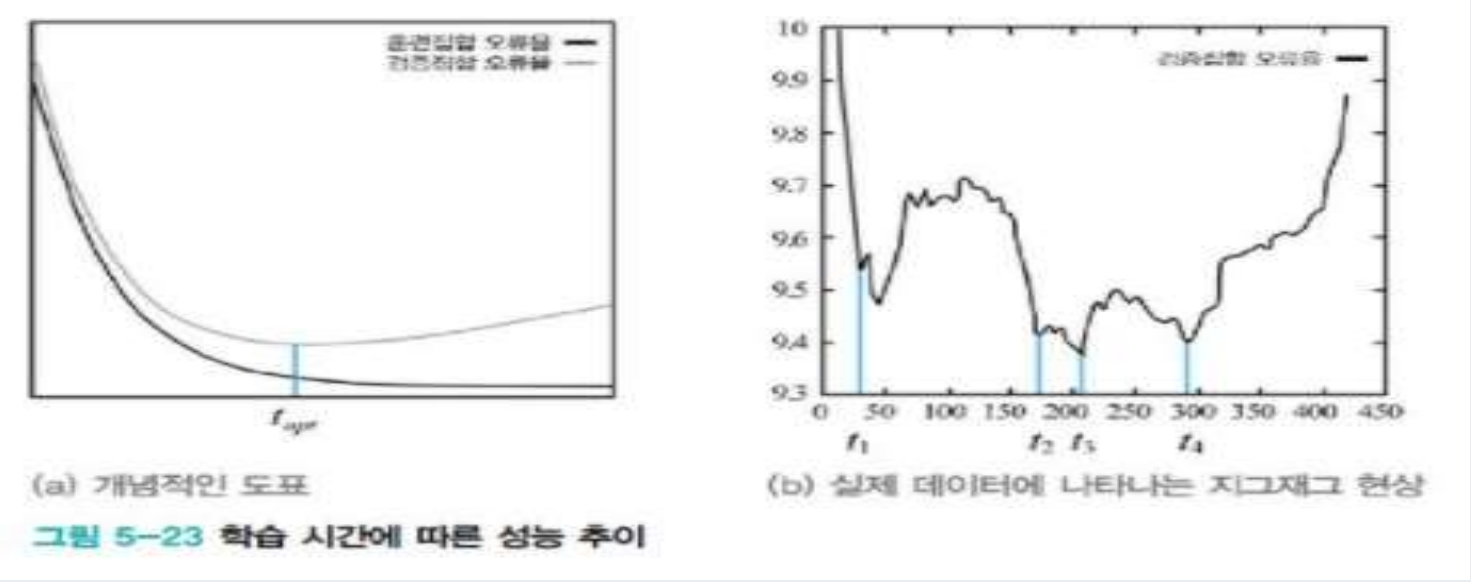
그림 5-22 *L1 놈을 사용한 가중치 감쇠 기법의 효과*

L1 놈은 0이 되는 매개변수가 많다는 현상을 입증하였고 이 현상을 희소성이라고 한다.
선형 회귀에 적용하면 0이 아닌 항만 남으므로 특징 선택 효과를 가져올 수 있다.

5.4.2 조기 멈춤

- 학습 시간에 따른 일반화 능력

일정시간이 지나면 과잉 적합 현상이 나타난다. -> 일반화 능력 저하
즉, 훈련 데이터를 단순히 암기하기 시작.



(a) 개념적인 도표 (b) 실제 데이터에 나타나는 지그재그 현상

그림 5-23 학습 시간에 따른 성능 추이

- 조기 멈춤이라는 규제 기법을 사용

검증집합의 오류가 최저인 점에서 학습을 멈춘다.

5.4.2 조기 멈춤

알고리즘 5-6 조기 멈춤을 채택한 기계 학습 알고리즘(지그재그 현상을 고려하지 않은 순진한 버전)

입력 : 훈련집합 $\mathbb{X}$와 $\mathbb{Y}$, 검증집합 $\mathbb{X}'$와 $\mathbb{Y}'$

출력 : 최적의 매개변수 $\hat{\Theta}$, 최적해가 발생한 세대 $\hat{t}$

```
1  난수를 생성하여 초기해 Θ0을 설정하고 오류율 e0 = 1.0으로 설정한다.   // 1.0은 오류율 최대치
2  t=0
3  while (true)
4     학습 알고리즘으로 Θt를 갱신하여 Θt+1을 얻는다.
5     Θt+1로 검증집합에 대한 오류율 et+1을 측정한다.
6     if(et+1 > et) break
7     t++
8  Θ̂ = Θt, t̂ = t
```

5.4.2 조기 멈춤

순진한 버전을 적용하면 t_1 에서 멈추므로 설익은 수렴이다.
이에 대처하는 여러 가지 방안 중에서 [알고리즘 5-7]은 참을성을 반영한 버전이다.

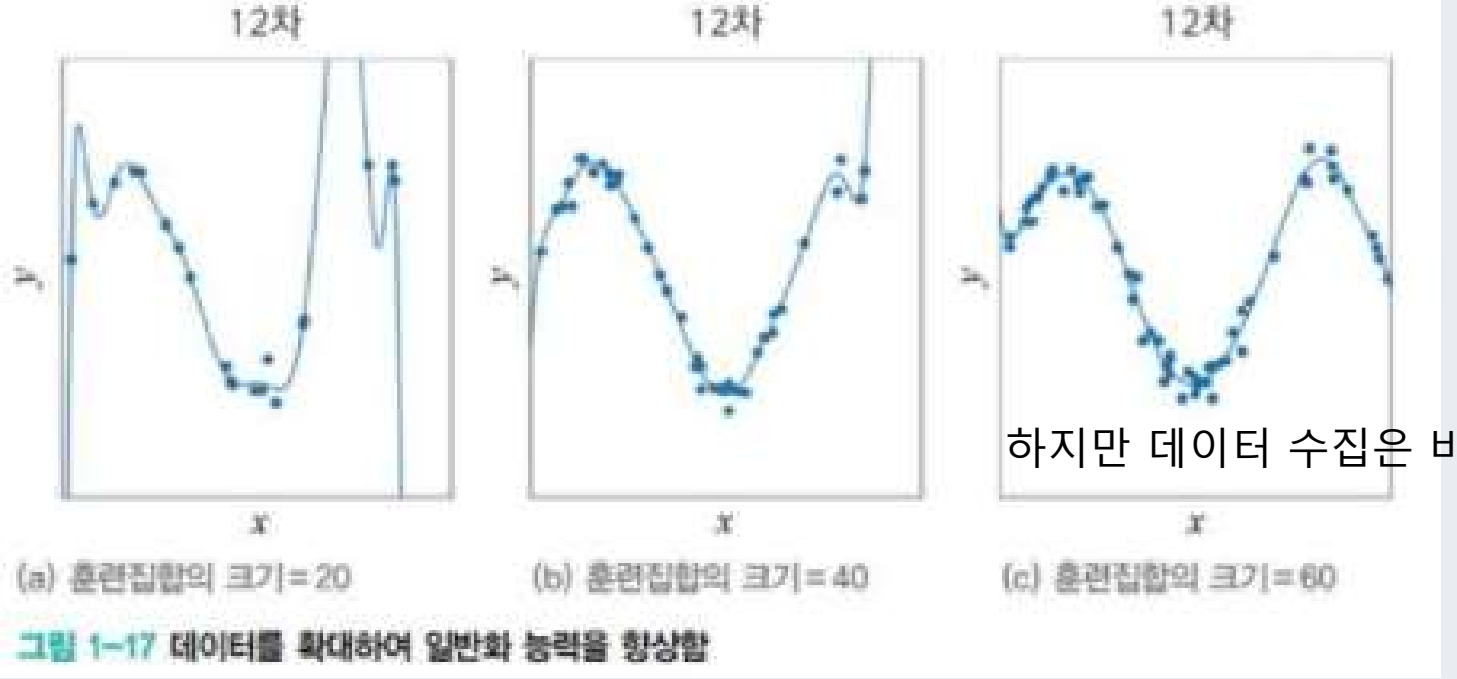
알고리즘 5-7 조기 멈춤을 채택한 기계 학습 알고리즘(참을성을 반영한 버전)

입력 : 훈련집합 $\mathbb{X}$와 $\mathbb{Y}$, 검증집합 $\mathbb{X}'$와 $\mathbb{Y}'$, 참을성 인자 p, 세대 반복 인자 q
출력 : 최적의 매개변수 $\hat{\Theta}$, 최적해가 발생한 세대 $\hat{t}$

```
1  난수를 생성하여 초기해 Θ0을 설정한다.
2  Θ̂ = Θ0, t̂ = 0
3  t = 0, ê = 1.0, j = 0
4  while (j<p)
5    학습 알고리즘의 세대를 q번 반복하여 Θt+q를 얻는다.
6    Θt+q로 검증집합에 대한 오류율 et+q를 측정한다.
7    if (et+q < ê) // 새로운 최적을 발견한 상황
8      j = 0   // 참는 과정을 처음부터 새로 시작
9      Θ̂ = Θt+q, ê = et+q, t̂ = t + q
10   else
11     j=j+1
12   t=t+q
```

5.4.3 데이터 확대

- 과잉적합 방지하는 가장 확실한 방법은 큰 훈련집합 사용



(a) 훈련집합의 크기=20 (b) 훈련집합의 크기=40 (c) 훈련집합의 크기=60

그림 1-17 데이터를 확대하여 일반화 능력을 향상함

하지만 데이터 수집은 비용이 많이 드는 작업이다.

- 데이터 확대라는 규제 기법을 적용

데이터를 인위적으로 변형하여 확대하였다.
자연계에서 벌어지는 잠재적인 변형을 프로그램으로 흉내 내어
샘플의 수를 강제로 늘리는 기법이라고 할 수 있다.

5.4.3 데이터 확대

예) MNIST에 어파인 변환(이동, 회전, 크기)을 적용

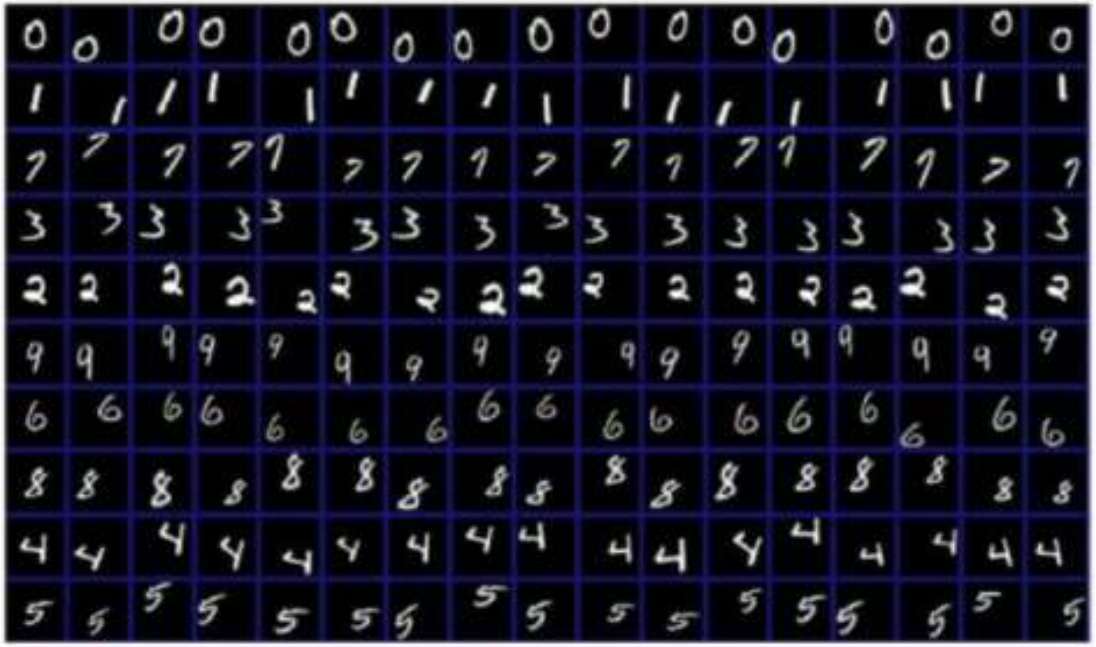


그림 5-24 필기 숫자 데이터의 다양한 변형

- 한계

수작업 변형
모든 부류가 같은 변형 사용

5.4.3 데이터 확대

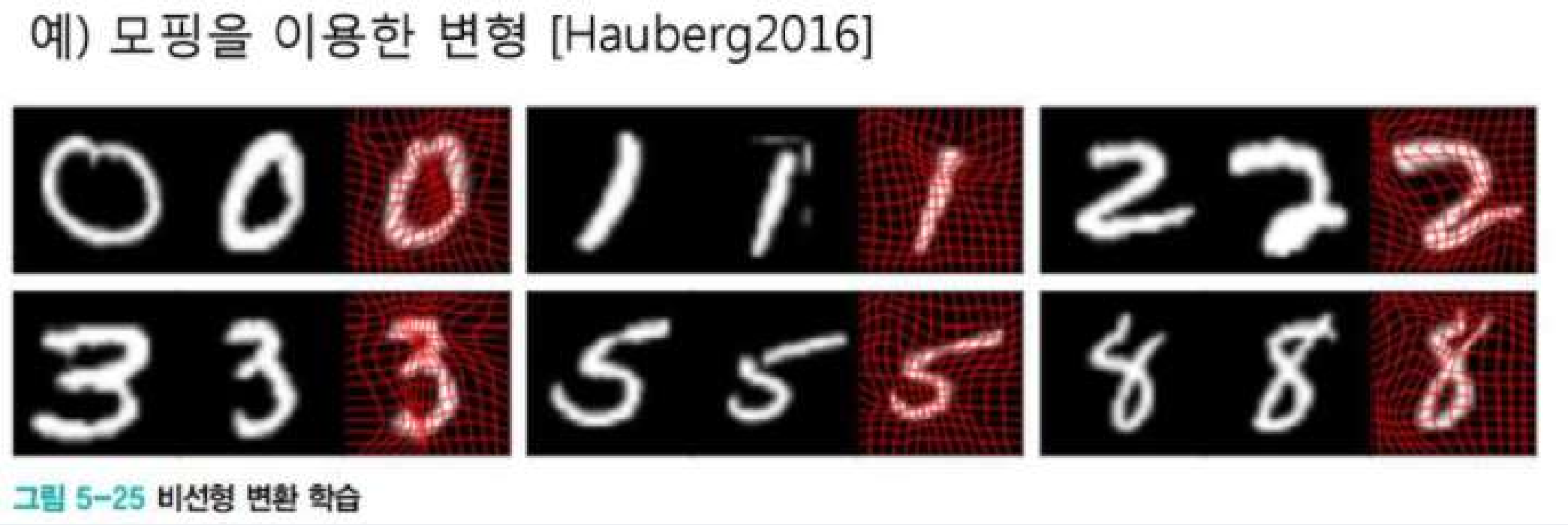


그림 5-25 비선형 변환 학습

비선형 변환으로 어파인 변환에 비해 훨씬 다양한 형태의 확대이다.
학습 기반 : 데이터에 맞는 비선형 변환 규칙을 학습 하는 셈이다.

5.4.3 데이터 확대

- 예) 자연영상 확대 [krizhevsky2012]
256*256 영상에서 224*224 영상을 1024장을 잘라내어 이동 효과 좌우 반전까지 시도하여 2048배로 확대
PCA를 이용한 색상 변환으로 추가 확대
예측 단계에서는 [그림 5-26]과 같이 5장을 잘라내고 좌우 반전하여 10장을 만든 다음 앙상블을 적용한다.

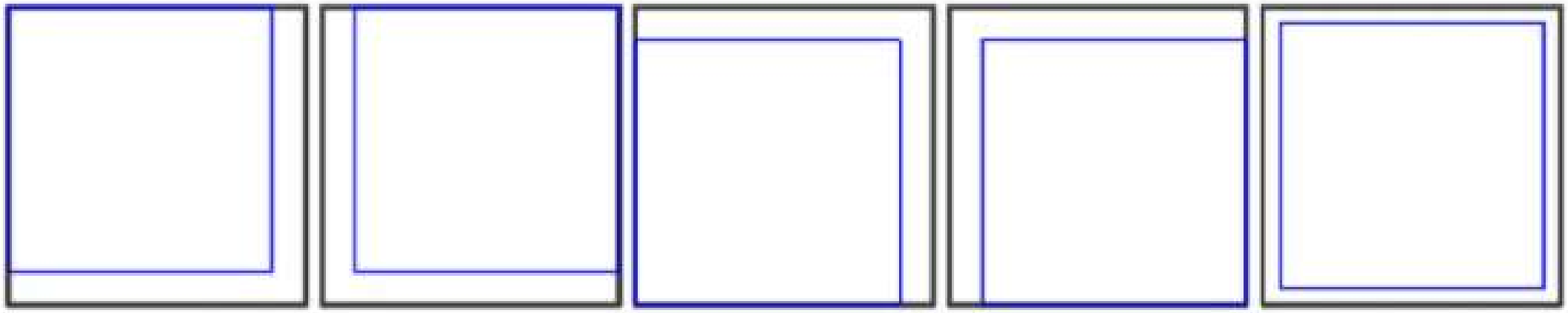


그림 5-26 예측 단계에서 영상 잘라내기

- 예) 잡음을 섞어 확대하는 기법
입력 데이터에 잡음을 섞는 기법이다.
은닉 노드에 잡음을 섞는 기법으로 고급 특징 수준에서 데이터를 확대하는 셈이다.

5.4.4 드롭아웃

- 드롭아웃 규제 기법

은닉층과 은닉층의 노드중 일정 비율을 임의로 선택하여 제거한다.
남은 부분 신경망을 학습한다.
많은 부분에 신경망을 만들고, 예측 단계에서 여러 개의 예측기를 결합하는 앙상블 기법의 일종으로 볼 수 있다.

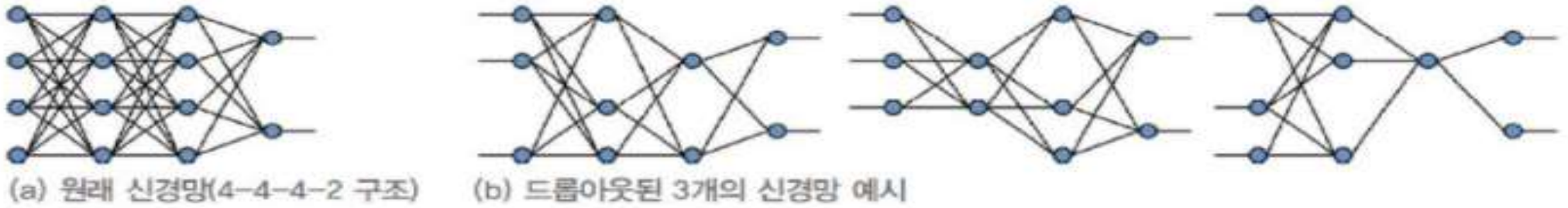


그림 5-27 드롭아웃된 신경망

많은 부분 신경망을 학습하고, 저장하고, 앙상블 결합하는 데 따른 계산 시간과 메모리 공간 측면의 부담한다.

드롭아웃 기법은 앙상블 효과를 거두면서 이러한 문제가 발생하지 않도록 조절하기 위해 하나의 신경망만 사용하는, 가중치를 공유하는 방식을 사용한다.
예측 단계에서도 하나의 신경망에 테스트 샘플을 입력하므로 계산 시간이 늘어나지 않는다.

5.4.4 드롭아웃

- 하나의 신경망(하나의 가중치 집합)에 드롭아웃을 적용하는 알고리즘

알고리즘 5-8 드롭아웃을 채택한 기계 학습 알고리즘

입력 : 드롭아웃 비율 p_{input}, p_{hidden}
출력 : 최적해 $\hat{\Theta}$

1 난수를 생성하여 초기해 Θ를 설정한다.
2 while (! 멈춤 조건) // 수렴 조건
3 미니배치 $\mathbb{B}$를 샘플링한다.
4 for (i=1 to $|\mathbb{B}|$) // $\mathbb{B}$의 샘플 각각에 대해
5 입력층은 p_{input}, 은닉층은 p_{hidden} 비율로 드롭아웃을 수행한다.
6 드롭아웃된 부분 신경망 $\Theta_i^{dropout}$로 전방 계산을 한다.
7 오류 역전파를 이용하여 $\Theta_i^{dropout}$를 위한 그레이디언트 $\nabla_i^{dropout}$를 구한다.
8 $\nabla_1^{dropout}, \nabla_2^{dropout}, \cdots, \nabla_{|\mathbb{B}|}^{dropout}$의 평균 $\nabla_{ave}^{dropout}$를 계산한다.
9 $\Theta = \Theta - \rho\nabla_{ave}^{dropout}$ // 가중치 갱신
10 $\hat{\Theta} = \Theta$

5.4.4 드롭아웃

- 라인 6
드롭아웃된 부분 신경망 $\theta_i^{dropout}$ 을 전방 계산한다.

l번째 은닉층의 j번째 노드의 연산:

$$z_j^l = \tau_l(s_j^l)$$

이때 $s_j^l = \mathbf{u}_j^l \mathbf{z}^{l-1}$

$$\Longrightarrow$$

드롭아웃 적용:

$$z_j^l = \tau_l(s_j^l)$$

이때 $\begin{cases} \tilde{\mathbf{z}}^{l-1} = \mathbf{z}^{l-1} \odot \boldsymbol{\pi}^{l-1} \\ s_j^l = \mathbf{u}_j^l \tilde{\mathbf{z}}^{l-1} \end{cases}$ (5.35)

불린 배열 π 에 노드 제거 여부를 표시한다.
π 는 샘플마다 독립적으로 정하는데, 난수를 이용하여 설정한다.
보통 입력층 제거 비율 $P_{input} = 0.2$, 은닉층 제거 비율 $P_{hidden} = 0.5$ 로 설정한다.

5.4.4 드롭아웃

- 예측 단계

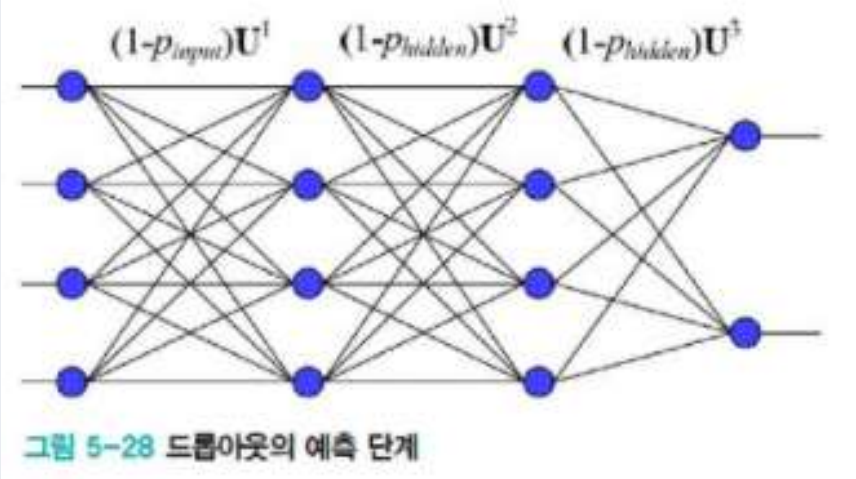


그림 5-28 드롭아웃의 예측 단계

예측 단계에서는 가중치 공유 기법을 통해 단 하나의
신경망을 만든다. 따라서 여러 개의 독립적인 신경망을 투표하여
의사결정하는 방식을 쓸 수 없고, 그림 5-28과 같이
단 하나의 신경망으로 앙상블 효과를 거두어야 한다.

- 앙상블 효과 모방

새로운 샘플이 입력되면 가중치에 생존 비율(1-드롭아웃 비율)을 곱하여 전방을 계산한다.
규모를 줄이는 이유는 학습 과정에서 가중치가 (1-드롭아웃 비율)만큼만 참여했기 때문이다.

-드롭 아웃의 장점 : 메모리와 계산 효율

추가 메모리는 불린 배열 π 만큼 필요하기 때문에 추가 계산은 작다.
실제 부담은 신경망의 크기에서 온다. 보통 은닉 노드의 수를 $1/P_{hidden}$ 만큼 늘린다.

5.4.5 앙상블 기법

- 앙상블

서로 다른 여러 개의 모델을 결합하여 일반화 오류를 줄이는 기법이다.
현대 기계 학습은 앙상블도 규제로 여긴다.

앙상블 기법에서는 서로 다른 예측기를 제작해야 한다.

1) 서로 다른 예측기를 학습하는 일
서로 다른 구조의 신경망을 여러 개 학습 또는 같은 구조를 사용하되 서로 다른 초깃값과 하이퍼 파라미터 매개변수를 설정하고 학습한다.
배깅 - 부트스트랩 기법을 앙상블 기법으로 확장한 것.
훈련 집합을 여러 번 샘플링하여 서로 다른 훈련집합을 구성한다.
부스팅 - i번째 예측기가 틀린 샘플을 i+1번째 예측기가 잘 인식하도록 연계성을 고려하여 앙상블을 구축한다.

2) 학습된 예측기를 결합하는 일
주로 투표 방식을 사용한다. => 모델 평균이라고 부른다.