2021 Spring

Artificial Intelligence & Deep Learning

Prof. Minsuk Koo

Department of Computer Science & Engineering
Incheon National University

5.3 규제의 필요성과 원리

- 5.3.1 과잉적합에 빠지는 이유와 과잉적합을 피하는 전략
- 5.3.2 규제의 정의

- 규제가 중요하기 때문에 1장에서 미리 소개한 내용
  - 1.5절의 과소적합과 과잉적합, 바이어스와 분산([그림 1-13], [그림 1-14])
  - 1.6절의 데이터 확대와 가중치 감쇠

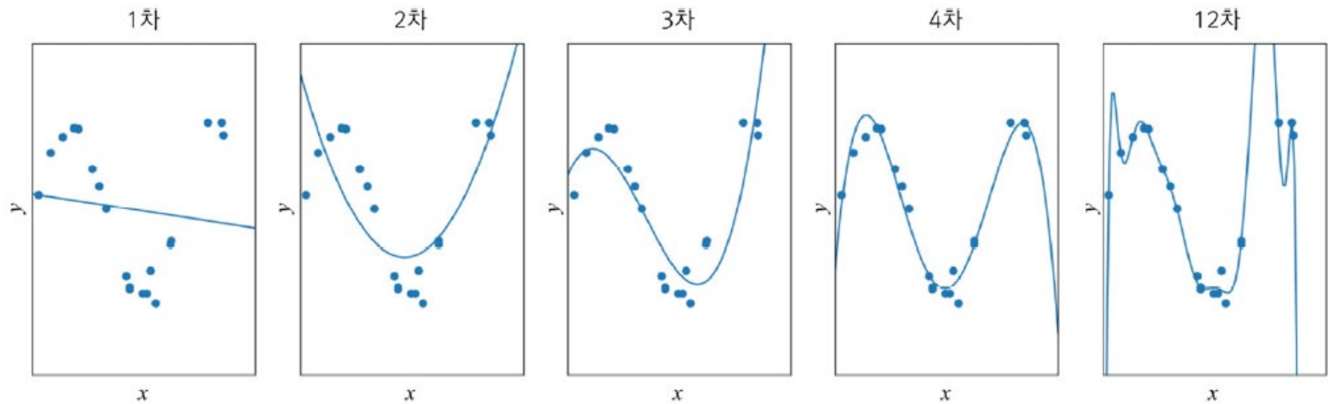


그림 1-13 과소적합과 과잉적합 현상

5.3.2 규제의 정의

- 현대 기계 학습도 매끄러움 가정을 널리 사용함
  - 5.4.1절의 가중치 감쇠 기법
    - 모델의 구조적 용량을 충분히 크게 하고, ‘수치적 용량’을 제한하는 규제 기법
  - 6장의 비지도 학습 등

- 『Deep Learning』책의 정의

“…any modification we make to a learning algorithm that is intended to reduce its generalization error … 일반화 오류를 줄이려는 의도를 가지고 학습 알고리즘을 수정하는 방법 모두”

5.4 규제 기법

- 5.4.1 가중치 벌칙
- 5.4.2 조기 멈춤
- 5.4.3 데이터 확대
- 5.4.4 드롭아웃
- 5.4.5 앙상블 기법

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

5.4.1 가중치 벌칙

- 식 (5.19)를 관련 변수가 드러나도록 다시 쓰면,

weight 함수

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

  - 규제항은 훈련집합과 무관하며, 데이터 생성 과정에 내재한 사전 지식에 해당
  - 규제항은 매개변수를 작은 값으로 유지하므로 모델의 용량을 제한하는 역할(수치적 용량을 제한함)

- 규제항 $R(\Theta)$로 무엇을 사용할 것인가?
  - 큰 가중치에 벌칙을 가해 작은 가중치를 유지하려고 주로 $L2$ 놈이나 $L1$ 놈을 사용

5.4.1 가중치 벌칙

$L2$ 놈

- 규제 항 R로 $L2$ 놈을 사용하는 규제 기법을 '가중치 감쇠'라weight decay 부름 → 식 (5.21)

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

L2 Norm

- 식 (5.21)의 그레이디언트 계산

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

5.4.1 가중치 벌칙

- 식 (5.22)를 이용하여 매개변수를 갱신하는 수식

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

- $\lambda = 0$으로 두면 규제를 적용하지 않은 원래 식 $\boldsymbol{\Theta} = \boldsymbol{\Theta} - \rho\nabla J$가 됨
- 가중치 감쇠는 단지 $\boldsymbol{\Theta}$에 $(1-2\rho\lambda)$를 곱해주는 셈
  - 예를 들어, ρ=0.01, $\lambda = 2.0$이라면 $(1-2\rho\lambda)$=0.96
- 최종해를 원점 가까이 당기는 효과 (즉 가중치를 작게 유지함)

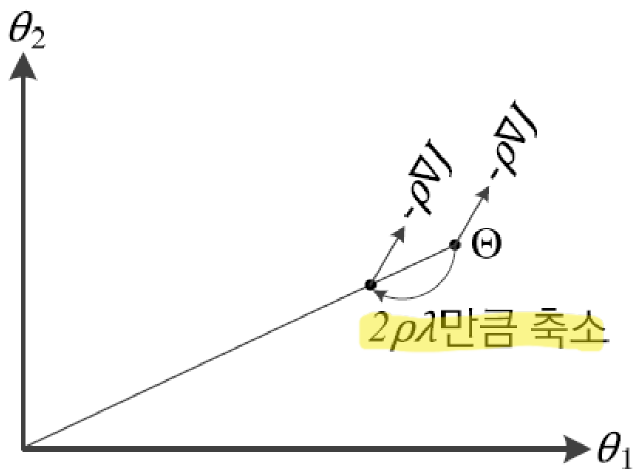


그림 5-21 $L2$ 놈을 사용한 가중치 감쇠 기법의 효과

5.4.1 가중치 벌칙

선형 회귀에 적용

- 선형 회귀는 훈련집합 $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, $\mathbb{Y} = \{y_1, y_2, \cdots, y_n\}$이 주어지면, 식 (5.24)를 풀어 $\mathbf{w} = (w_1, w_2, \cdots, w_d)^{\mathrm{T}}$를 구하는 문제. 이때 $\mathbf{x}_i = (x_{i1}, x_{i2}, \cdots, x_{id})^{\mathrm{T}}$

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

- 식 (5.24)를 행렬식으로 바꿔 쓰면,

$$\mathbf{X}\mathbf{w} = \mathbf{y} \tag{5.25}$$

- 가중치 감쇠를 적용한 목적함수

$$J_{regularized}(\mathbf{w}) = \underbrace{\frac{1}{2}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_2^2}_{\text{MSE}} = \frac{1}{2}(\mathbf{X}\mathbf{w} - \mathbf{y})^{\mathrm{T}}(\mathbf{X}\mathbf{w} - \mathbf{y}) + \lambda\|\mathbf{w}\|_2^2 \tag{5.27}$$

5.4.1 가중치 벌칙

- 식 (5.27)을 미분하여 0으로 놓으면,

$$\frac{\partial J_{regularized}}{\partial \mathbf{w}} = \mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w} - \mathbf{X}^{\mathrm{T}}\mathbf{y} + 2\lambda\mathbf{w} = \mathbf{0} \implies (\mathbf{X}^{\mathrm{T}}\mathbf{X} + 2\underset{\text{scalar}}{\lambda}\overset{\begin{pmatrix}1&0\\0&1\end{pmatrix}}{\mathbf{I}})\mathbf{w} = \mathbf{X}^{\mathrm{T}}\mathbf{y} \tag{5.28}$$

- 식 (5.28)을 정리하면,

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

  - 공분산 행렬 $\mathbf{X}^{\mathrm{T}}\mathbf{X}$의 대각 요소가 2λ만큼씩 증가 → 역행렬을 곱하므로 가중치를 축소하여 원점으로 당기는 효과 ([그림 5-21])

- 예측 단계에서는.

$$y = \mathbf{x}^{\mathrm{T}}\hat{\mathbf{w}} \tag{5.30}$$

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.

x_1^T

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

x_3^T

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

$\hookrightarrow \hat{w} = (X^TX + 2\lambda I)^{-1} X^TY$

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

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

5.4.1 가중치 벌칙

MLP와 DMLP에 적용

- 식 (3.21)에 식 (5.23)의 가중치 감쇠라는 규제 기법을 적용하면,

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

- [알고리즘 3-4]에 적용하면,

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

$\Downarrow$

13. for (k=1 to c) for (j=0 to p) $u_{kj}^2 = (1-2\rho\lambda)u_{kj}^2 - \rho\Delta u_{kj}^2$ // 가중치 감쇠 적용한 알고리즘
14. for (j=1 to p) 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](미니배치 버전)에 적용하면,

14. $\mathbf{U}^2 = \mathbf{U}^2 - \rho\frac{\Delta\mathbf{U}^2}{t}$ // 가중치 감쇠 적용하지 않은 원래 알고리즘
15. $\mathbf{U}^1 = \mathbf{U}^1 - \rho\frac{\Delta\mathbf{U}^1}{t}$

$\Downarrow$

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}) $u_{ji}^l = u_{ji}^l - \rho\left(\frac{1}{t}\right)\Delta u_{ji}^l$

$\Downarrow$

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 $\|\boldsymbol{\Theta}\|_1 = |\theta_1| + |\theta_2| + \cdots$)

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

- 식 (5.32)를 미분하면,

$$\nabla J_{regularized}(\boldsymbol{\Theta}; \mathbb{X}, \mathbb{Y}) = \nabla J(\boldsymbol{\Theta}; \mathbb{X}, \mathbb{Y}) + \lambda \underbrace{\mathbf{sign}}_{+ \text{ or } -}(\boldsymbol{\Theta}) \quad (5.33)$$

- 매개변수를 갱신하는 식에 대입하면,

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

단순히 부호에만 영향

5.4.1 가중치 벌칙

- 매개변수를 갱신하는 식

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

- 식 (5.34)의 가중치 감쇠 효과

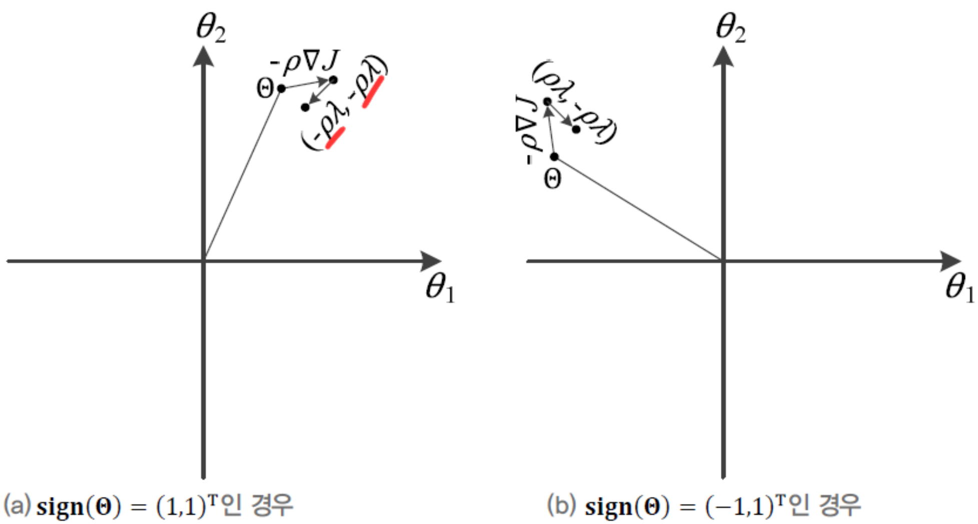
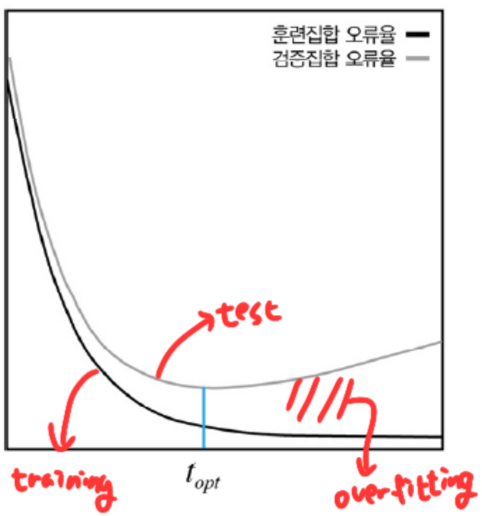


그림 5-22 $L1$ 놈을 사용한 가중치 감쇠 기법의 효과

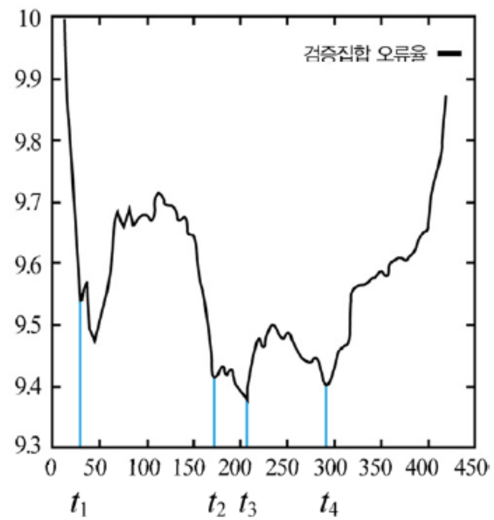
- $L1$ 놈의 희소성 효과(0이 되는 매개변수가 많음)
  - 선형 회귀에 적용하면 특징 선택 효과

5.4.2 조기 멈춤

- 학습 시간에 따른 일반화 능력 [그림 5-23(a)]
  - 일정 시간(t_{opt})이 지나면 과잉적합 현상이 나타남 → 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

= early stopping

- 조기 멈춤이라는 규제 기법
  - 검증집합의 오류가 최저인 점 t_{opt}에서 학습을 멈춤

5.4.2 조기 멈춤

- [알고리즘 5-6]은 현실을 제대로 반영하지 않은 순진한 버전
  - [그림 5-23(a)] 상황에서 동작

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 $\widehat{\Theta}$, 최적해가 발생한 세대 $\hat{t}$

난수를 생성하여 초기해 Θ_0을 설정하고 오류율 $e_0 = 1.0$으로 설정한다. // 1.0은 오류율 최대치

$t=0$

while (true)

  학습 알고리즘으로 Θ_t를 갱신하여 Θ_{t+1}을 얻는다.

  Θ_{t+1}로 검증집합에 대한 오류율 e_{t+1}을 측정한다.

  if($e_{t+1} > e_t$) break

  t++

$\widehat{\Theta} = \Theta_t$, $\hat{t} = t$

5.4.2 조기 멈춤

- 실제 세계는 [그림 5-23(b)]와 같은 상황
  - 순진한 버전을 적용하면 t_1에서 멈추므로 설익은 수렴
  - 이에 대처하는 여러 가지 방안 중에서 [알고리즘 5-7]은 참을성을 반영한 버전

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 $\widehat{\Theta}$, 최적해가 발생한 세대 $\hat{t}$

p만큼 기다림

난수를 생성하여 초기해 Θ_0을 설정한다.
$\widehat{\Theta} = \Theta_0,\ \hat{t} = 0$
$t = 0,\ \hat{e} = 1.0,\ j = 0$
while $(j < p)$
    학습 알고리즘의 세대를 q번 반복하여 Θ_{t+q}를 얻는다.
    Θ_{t+q}로 검증집합에 대한 오류율 e_{t+q}를 측정한다.
    if $(e_{t+q} < \hat{e})$ // 새로운 최적을 발견한 상황
        $j = 0$ // 참는 과정을 처음부터 새로 시작
        $\widehat{\Theta} = \Theta_{t+q},\ \hat{e} = e_{t+q},\ \hat{t} = t + q$
    else
        $j = j + 1$
    $t = t + q$

5.4.3 데이터 확대

- 과잉적합 방지하는 가장 확실한 방법은 큰 훈련집합 사용

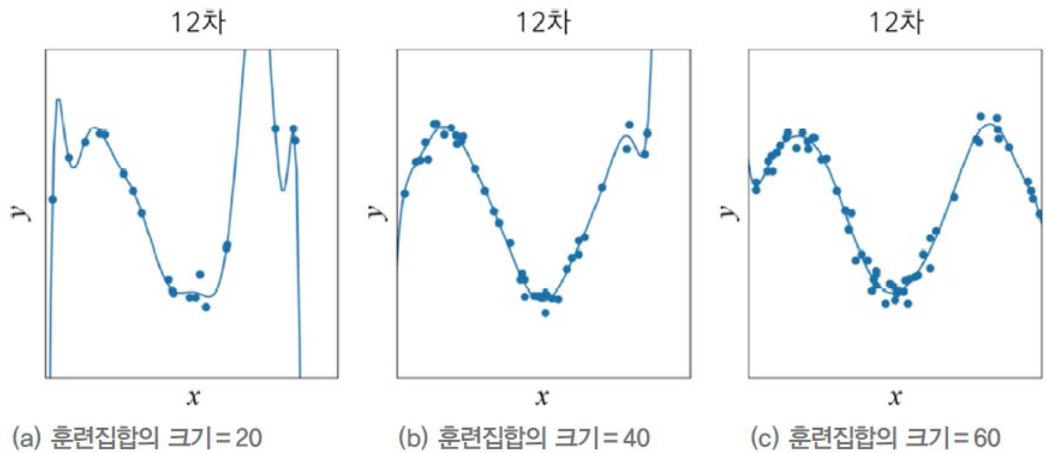


그림 1-17 데이터를 확대하여 일반화 능력을 향상함

  - 하지만 데이터 수집은 비용이 많이 드는 작업

- 데이터 확대라는 규제 기법
  - 데이터를 인위적으로 변형하여 확대함
  - 자연계에서 벌어지는 잠재적인 변형을 프로그램으로 흉내 내는 셈

5.4.3 데이터 확대

- 예) MNIST에 어파인 변환(이동, 회전, 크기)을 적용

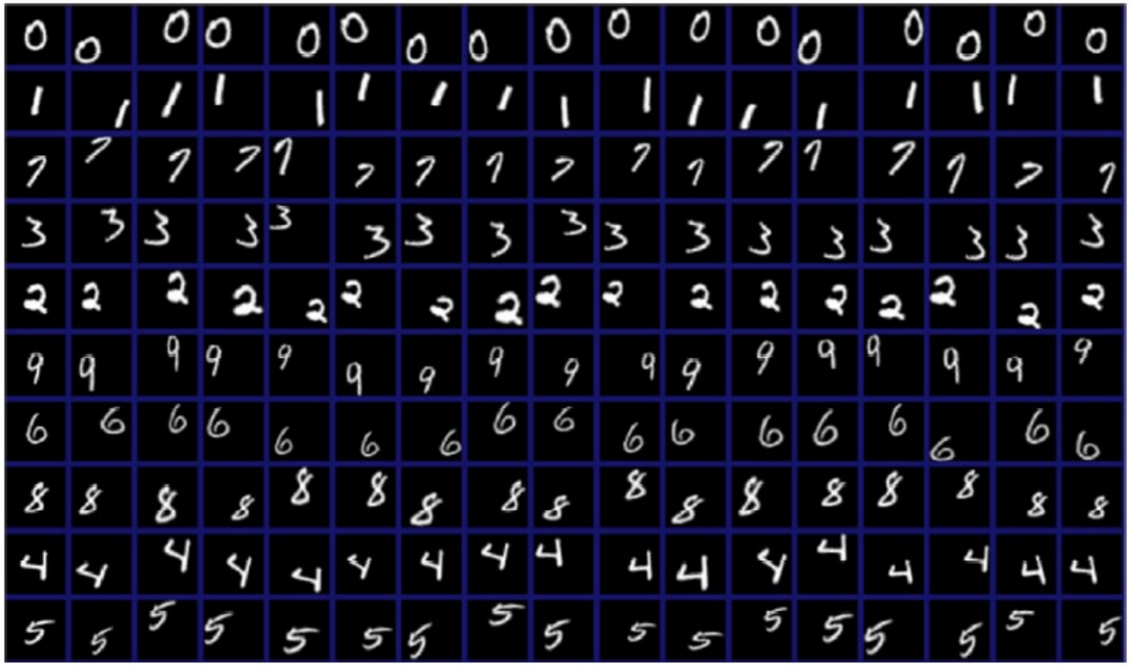


그림 5-24 필기 숫자 데이터의 다양한 변형[8]

  - 한계
    - 수작업 변형
    - 모든 부류가 같은 변형 사용

5.4.3 데이터 확대

- 예) 모핑을 이용한 변형 [Hauberg2016]

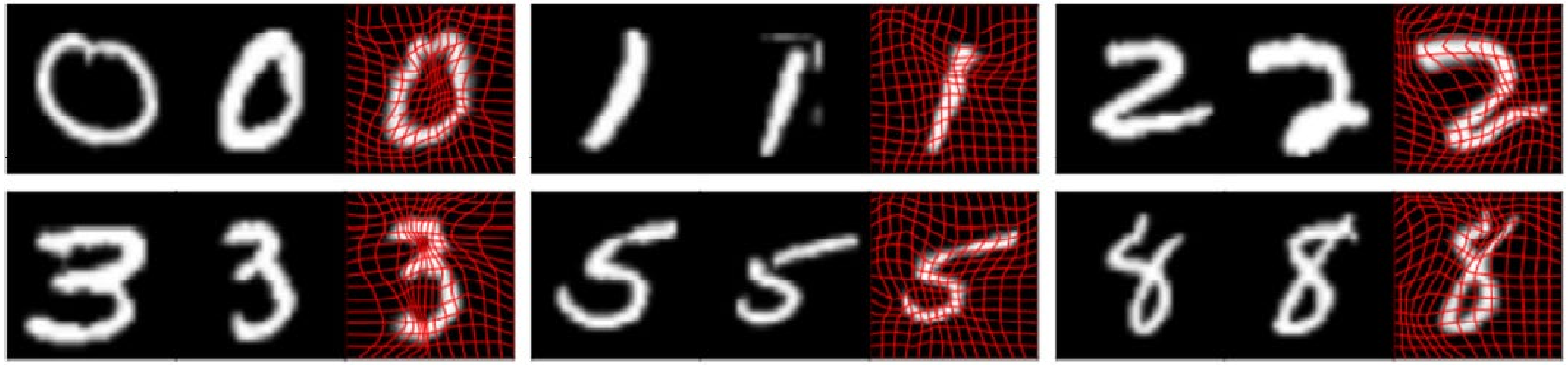


그림 5-25 비선형 변환 학습

- 비선형 변환으로서 어파인 변환에 비해 훨씬 다양한 형태의 확대
- 학습 기반: 데이터에 맞는 '비선형 변환 규칙을 학습'하는 셈

5.4.3 데이터 확대

- 예) 자연영상 확대 [Krizhevsky2012]
  - 256*256 영상에서 224*224 영상을 1024장 잘라내어 이동 효과. 좌우 반전까지 시도하여 2048배로 확대
  - PCA를 이용한 색상 변환으로 추가 확대
  - 예측 단계에서는 [그림 5-26]과 같이 5장 잘라내고 좌우 반전하여 10장을 만든 다음 앙상블 적용

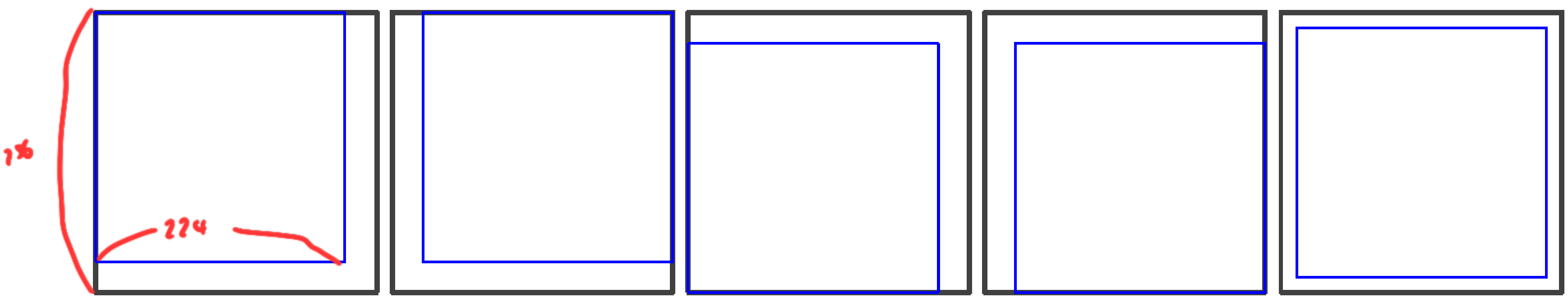


그림 5-26 예측 단계에서 영상 잘라내기

- 예) 잡음을 섞어 확대하는 기법
  - 입력 데이터에 잡음을 섞는 기법
  - 은닉 노드에 잡음을 섞는 기법 (고급 특징 수준에서 데이터를 확대하는 셈)