인 공 지 능

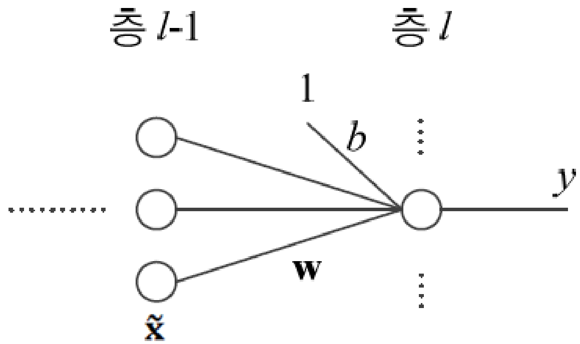
[심층학습 최적화 III]

jaekoo@kookmin.ac.kr

국민대학교
KOOKMIN UNIVERSITY

5.2.5 활성함수

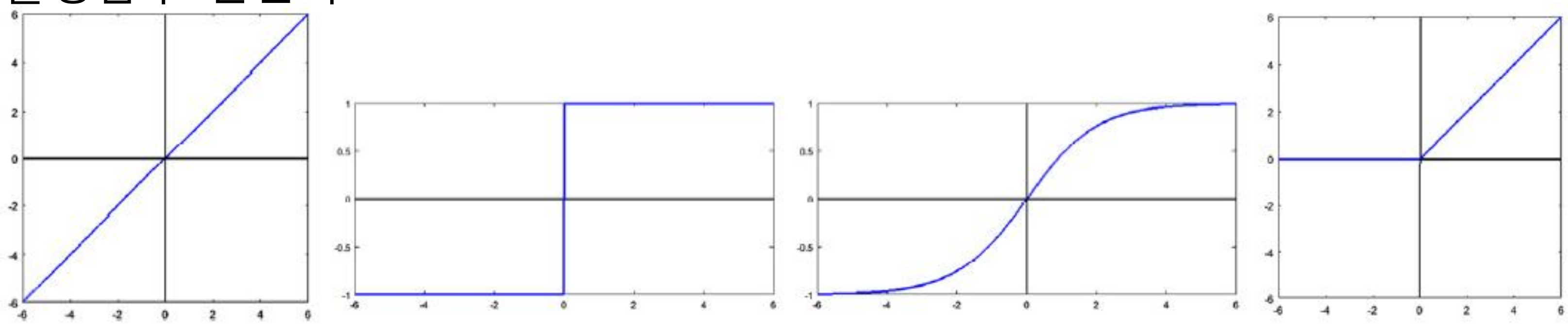
- 선형 연산 결과인 활성값 z에 비선형 활성함수 τ를 적용하는 과정



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

그림 5-14 신경망 노드의 연산

- 활성함수 변천사



(a) 선형 (b) 계단(1950년대) (c) tanh(1980년대) (d) ReLU(2000년경~현재)

그림 5-15 활성함수 τ

- sigmoid 함수는 활성값이 커지면 포화 상태가 되고 경사도가 0에 가까운 값을 출력함

→ 매개변수 갱신 (학습)이 매우 느린 요인

5.2.5 활성함수

- ReLU (Rectified Linear Unit) 활성함수
  - 경사도 포화gradient saturation 문제 해소

$$z = \mathbf{w}^{\mathrm{T}}\tilde{\mathbf{x}} + b$$
$$y = \mathrm{ReLU}(z) = \max(0, z) \qquad (5.16)$$

- ReLU의 변형
  - Leaky ReLU (보통 $\alpha = 0.01$을 사용) $\mathrm{leakyReLU}(z) = \begin{cases} z, & z \geq 0 \\ \alpha z, & z < 0 \end{cases}$ (5.17)
  - Parametric ReLU (α를 학습으로 알아냄)

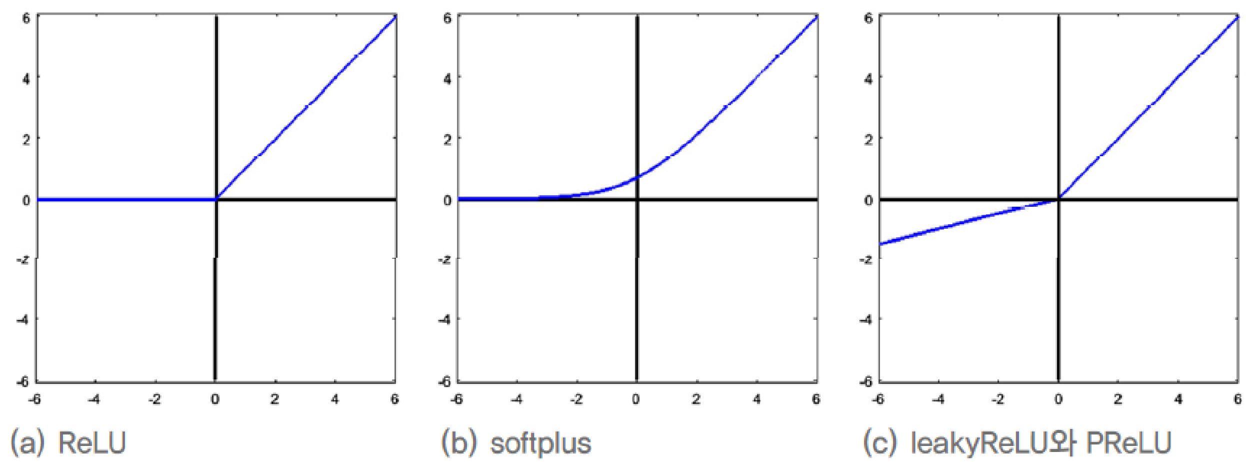


그림 5-16 ReLU의 변형

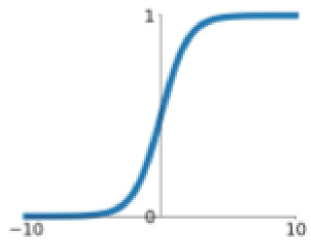
국민대학교 KOOKMIN UNIVERSITY

5.2.5 활성함수

- 다양한 활성함수들

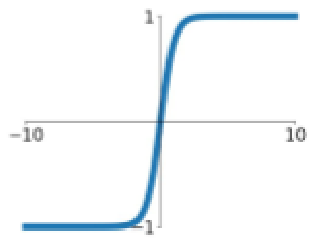
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



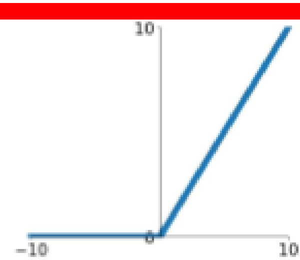
tanh

$\tanh(x)$



ReLU

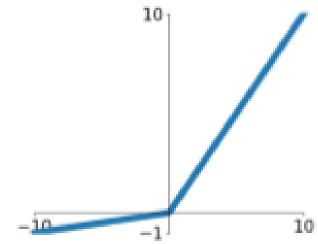
$\max(0, x)$



Good default choice

Leaky ReLU

$\max(0.1x, x)$

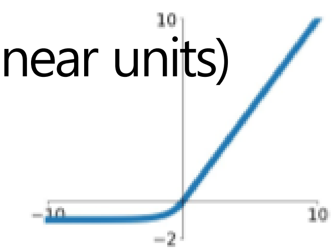


Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU (exponential linear units)

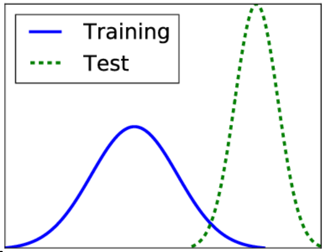
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



최근의 활성 함수들은 다음의 문제들을 해결하고자 함
- 포화된 영역이 경사도가 작아짐
- 출력값이 영 중심 아님
- 다소 높은 연산량 (e.g., Exp() 함수)

5.2.6 배치 정규화

- 공변량 변화covariate shift 현상
  - 훈련집합과 테스트집합의 분포가 다름



  - 내부의 공변량 변화internal covariate shift
    - 학습이 진행되면서 첫번째 층의 매개변수가 바뀜에 따라 $\tilde{\mathbb{x}}^{(1)}$이 따라 바뀜

      → 두번째 층 입장에서 보면 자신에게 입력되는 데이터의 분포가 수시로 바뀌는 셈
    - 층2, 층3, …으로 깊어짐에 따라 더욱 심각→ 학습을 방해하는 요인으로 작용

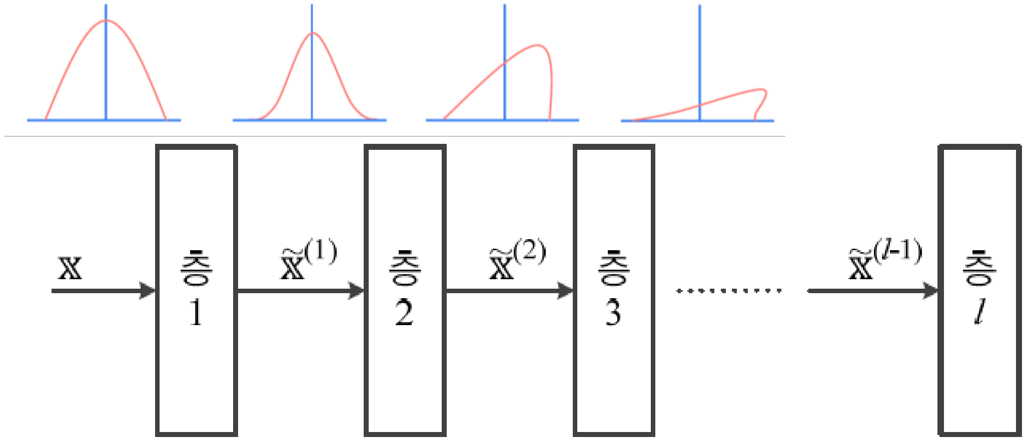


그림 5-17 공변량 시프트 현상

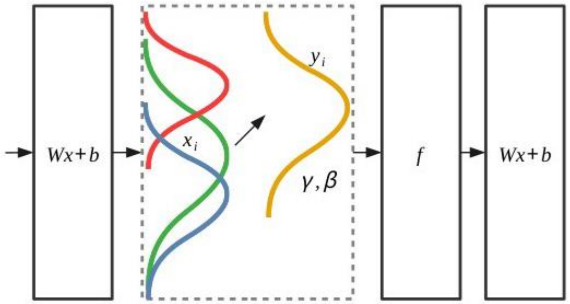
5.2.6 배치 정규화

- 배치 정규화batch normalization
  - 공변량 시프트 현상을 누그러뜨리기 위해 식 (5.9)의 정규화를 층 단위 적용하는 기법

$$x_i^{new} = \frac{x_i^{old} - \mu_i}{\sigma_i} \qquad (5.9)$$

  - 정규화를 적용하는 곳이 중요
    - 식 (5.15)의 연산 과정 중 식 (5.9)를 어디에 적용하나? (적용 위치)

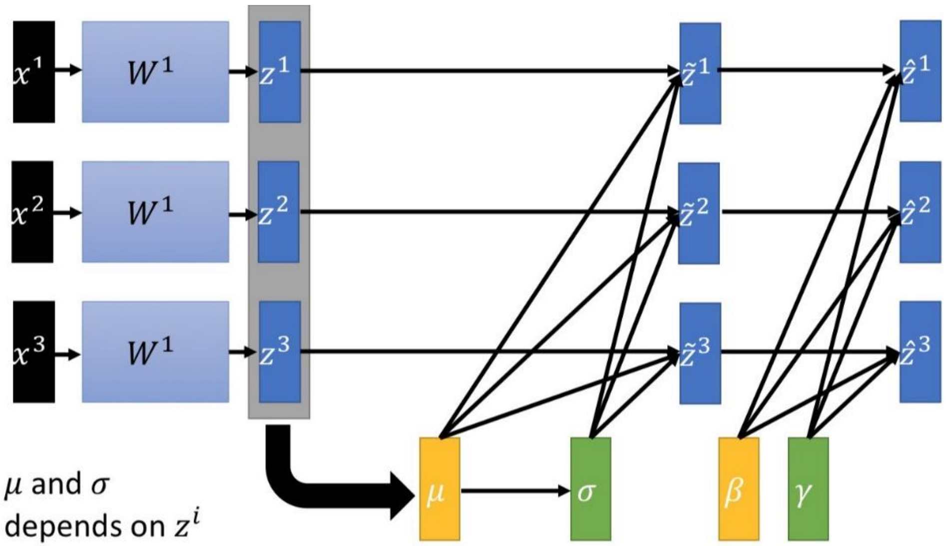
$$z = \mathbf{w}^{\mathrm{T}}\tilde{\mathbf{x}} + b$$
$$y = \tau(z) \qquad (5.15)$$



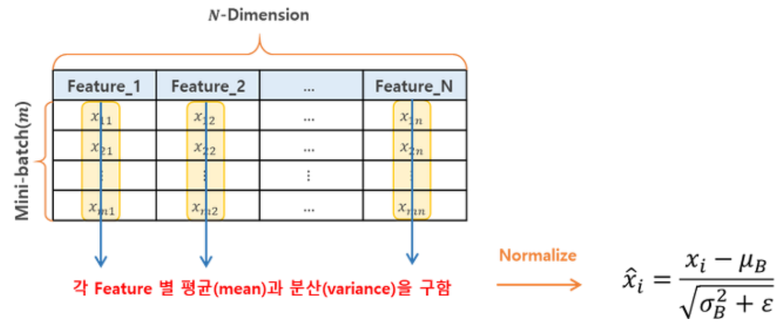
    - 입력 $\tilde{\mathbf{x}}$ 또는 중간 결과 z 중 어느 것에 적용? → z에 적용하는 것이 유리
    - 일반적으로 완전연결층, 합성곱층 후 혹은 비선형 함수 전 적용
  - 훈련집합 전체 또는 미니배치 중 어느 것에 적용? (적용 단위)
    - 미니배치에 적용하는 것이 유리

5.2.6 배치 정규화

배치 정규화 과정



1. 미니배치 단위로 평균(μ)과 분산(σ) 계산
2. 구한 평균과 분산을 통해 정규화



3. 비례(γ)와 이동(β) 세부 조정

배치 정규화 장점

- 신경망의 경사도 흐름 개선
- 높은 학습률 허용
- 초기화에 대한 의존성 감소
- 의도하지 않았지만 규제와 유사한 행동을 하며, 드롭아웃의 필요성을 감소시킴

5.2.6 배치 정규화

정규화 변환을 수행하는 코드

- 미니배치 $\mathbb{X}_B = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\}$에 식 (5.15)를 적용하여 $\tilde{\mathbb{X}}_B = \{z_1, z_2, \cdots, z_m\}$를 얻은 후, $\tilde{\mathbb{X}}_B$를 가지고 코드 1을 수행
- 즉, 미니배치 단위로 노드마다 독립적으로 코드 1을 수행
- γ와 β는 노드마다 고유한 매개변수로서 학습으로 알아냄

코드 1:

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} z_i$$ # 미니배치 평균

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m} (z_i - \mu_B)^2$$ # 미니배치 분산

$$\tilde{z}_i = \frac{z_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}, \qquad i = 1,2,\cdots,m$$ # 정규화

$$z_i' = \gamma \tilde{z}_i + \beta, \qquad i = 1,2,\cdots,m$$ # 비례scale와 이동shift

(β가 편향 역할을 하므로 식 (5.15)의 가중치 편향 b는 제거해도 됨)

5.2.6 배치 정규화

- 최적화를 마친 후 추가적인 후처리 작업 필요
  - 각 노드는 전체 훈련집합을 가지고 독립적으로 코드2를 수행

코드 2:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} z_i$$

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (z_i - \mu)^2$$

노드에 μ, σ^2, γ, β를 저장한다. // 예측 단계에서 식 (5.18)로 변환을 수행하기 위함

- 예측 단계
  - 각 노드는 독립적으로 식 (5.18)을 적용하여 변환 (코드 1의 마지막 두 라인을 수행하는 …)

$$z' = \frac{\gamma}{\sqrt{\sigma^2 + \varepsilon}} z + \left(\beta - \frac{\gamma\mu}{\sqrt{\sigma^2 + \varepsilon}}\right) \quad (5.18)$$

5.2.6 배치 정규화

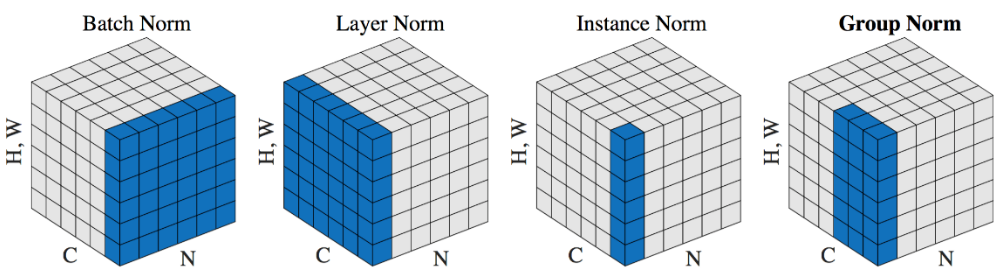
- CNN에서는 노드 단위가 아니라 특징 맵 단위로 코드 1과 코드 2를 적용
  - 예를 들면, 특징 맵의 크기가 p*q라면 미니배치에 있는 샘플마다 pq개의 값이 발생

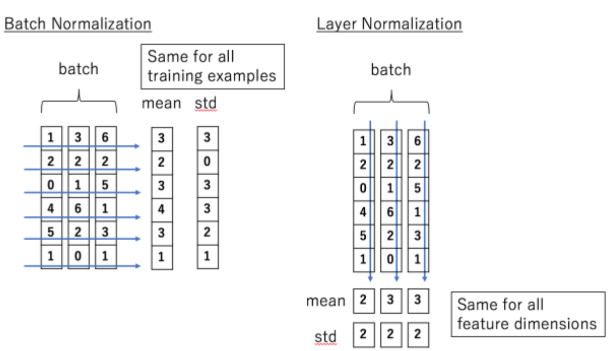
    코드 1은 총 pqm개의 값을 가지고 μ_B와 σ_B^2를 계산

    γ와 β는 특징 맵마다 하나씩 존재

- 배치 정규화의 긍정적 효과를 측정한 실험사례 [Ioffe2015]
  - 가중치 초기화에 덜 민감함
  - 학습률을 크게 하여 수렴 속도 향상 가능
  - sigmoid 활성함수로 사용하는 깊은 신경망도 학습이 이루어짐
  - 규제 효과를 제공하여 드롭아웃을 적용하지 않아도 높은 성능

- 다양한 정규화 방법들





5.3 규제의 필요성과 원리

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

5.3.1 과잉적합에 빠지는 이유와 과잉적합을 피하는 전략

- 학습 모델의 용량에 따른 일반화 능력

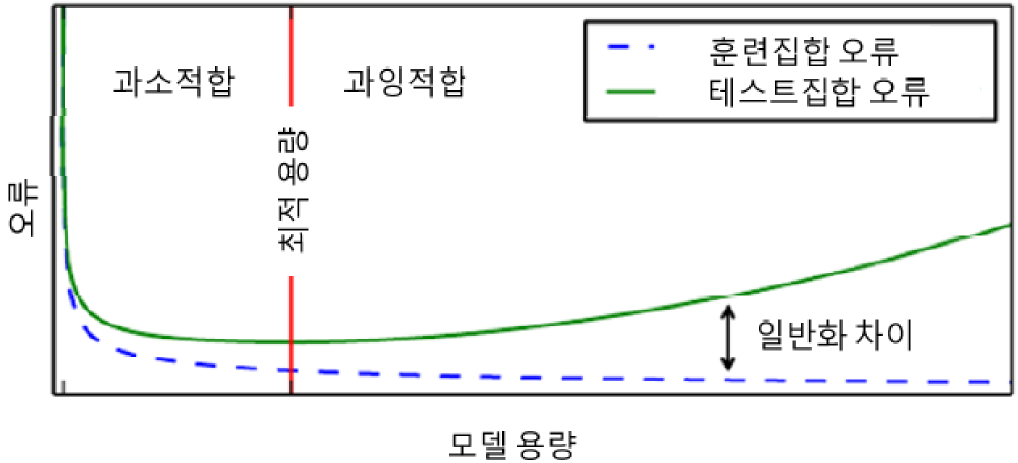


그림 5-18 학습 모델의 용량과 일반화 능력의 관계

- 대부분 가지고 있는 데이터에 비해 훨씬 큰 용량의 모델을 사용
  - 예) VGGNet은 분류층(완전연결층)에 1억 2천 1백만 개의 매개변수
  - 훈련집합을 단순히 '암기'하는 과잉적합에 주의를 기울여야 함
- 현대 기계학습의 전략
  - 충분히 큰 용량의 모델을 설계한 다음, 학습 과정에서 여러 규제 기법을 적용

5.3.2 규제의 정의

- 규제는 오래 전부터 수학과 통계학에서 연구해온 주제
  - 모델 용량에 비해 데이터가 부족한 경우의 부족조건문제를ill-posed problem 푸는 접근법
  - 적절한 가정을 투입하여 문제를 품 → '입력과 출력 사이의 변환은 매끄럽다'는 사전 지식
    - 유사한 데이터는 가깝게 매핑 된다.

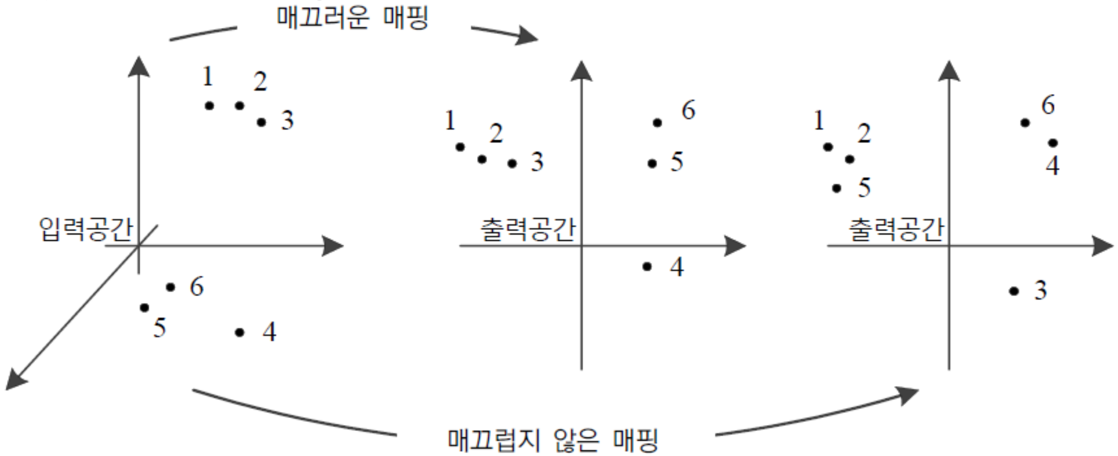


그림 5-20 사전 지식으로서 매끄러움의 특성

  - 대표적인 티호노프의 규제Tikhonov's regularization 기법는 매끄러움 가정에 기반을 둔 식 (5.19)를 사용
    - 통계에서는 릿지 회귀ridge regression, 기계학습에서는 가중치 감쇄weight decay 등이 대표적임

$$A\mathbf{x} = \mathbf{b}$$

$$\|A\mathbf{x} - \mathbf{b}\|_2^2$$

$$\|A\mathbf{x} - \mathbf{b}\|_2^2 + \|\Gamma\mathbf{x}\|_2^2$$

$$\underbrace{J_{regularized}(\Theta)}_{\text{규제를 적용한 목적함수}} = \underbrace{J(\Theta)}_{\text{목적함수}} + \lambda \underbrace{R(\Theta)}_{\text{규제 항}} \qquad (5.19)$$

5.3.2 규제의 정의

- 현대 기계학습도 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,

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

  - 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

- $L2$ 놈norm
  - 규제 항 R로 $L2$ 놈을 사용하는 규제 기법을 '가중치 감쇠'weight decay라 부름 → 식 (5.21)

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

  - 식 (5.21)의 경사도 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

$$\mathbf{\Theta} = \mathbf{\Theta} - \rho\nabla J_{regularized}(\mathbf{\Theta}; \mathbb{X}, \mathbb{Y})$$
$$= \mathbf{\Theta} - \rho(\nabla J(\mathbf{\Theta}; \mathbb{X}, \mathbb{Y}) + 2\lambda\mathbf{\Theta})$$
$$= (1 - 2\rho\lambda)\mathbf{\Theta} - \rho\nabla J(\mathbf{\Theta}; \mathbb{X}, \mathbb{Y}) \longrightarrow \underline{\mathbf{\Theta} = (1 - 2\rho\lambda)\mathbf{\Theta} - \rho\nabla J} \quad (5.23)$$

← $\lambda = 0$으로 두면 규제를 적용하지 않은 원래 식 $\mathbf{\Theta} = \mathbf{\Theta} - \rho\nabla J$가 됨

- 가중치 감쇠는 단지 $\mathbf{\Theta}$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)

= 가중치 감쇠^{decay} 효과

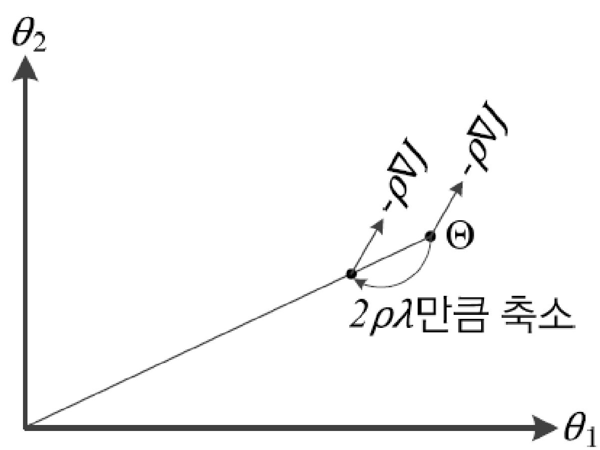
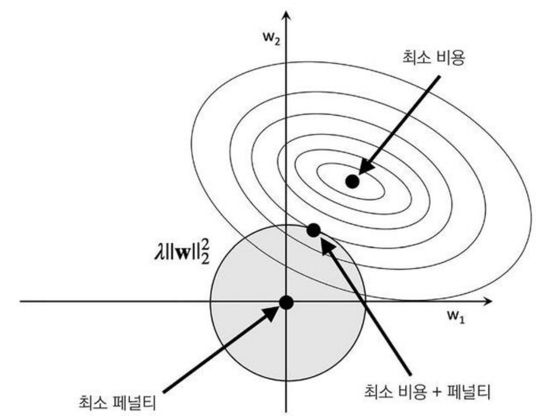


그림 5-21 $L2$ 놈을 사용한 가중치 감쇠 기법의 효과

최적화 해석



5.4.1 가중치 벌칙

선형 회귀linear regression에 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

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

  - L1 규제 – Lasso regression
  - L2 규제 – Ridge regression

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,

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

  - 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

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

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 $\mathbf{x} = (5 \quad 4)^{\mathrm{T}}$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,

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

- [알고리즘 3-4]에 적용하면,

13. for (k=1 to c) for (j=0 to 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 버전에 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,

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

- 매개변수를 갱신하는 식에 대입하면,

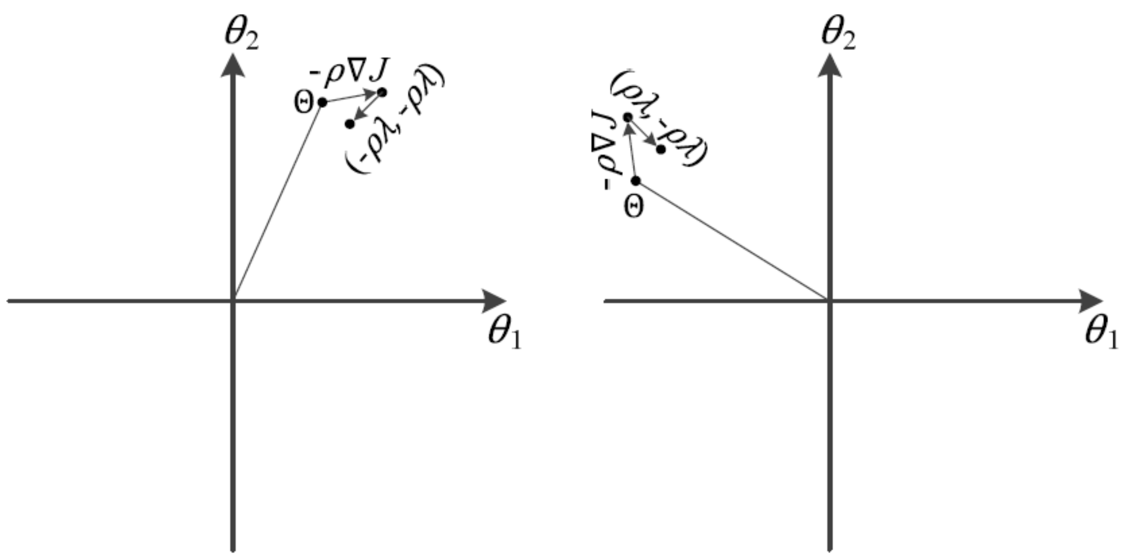
$$\begin{aligned}\boldsymbol{\Theta} &= \boldsymbol{\Theta} - \rho \nabla J_{regularized}(\boldsymbol{\Theta}; \mathbb{X}, \mathbb{Y}) \\ &= \boldsymbol{\Theta} - \rho(\nabla J(\boldsymbol{\Theta}; \mathbb{X}, \mathbb{Y}) + \lambda \mathbf{sign}(\boldsymbol{\Theta})) \\ &= \boldsymbol{\Theta} - \rho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

5.4.1 가중치 벌칙

- 매개변수를 갱신하는 식

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

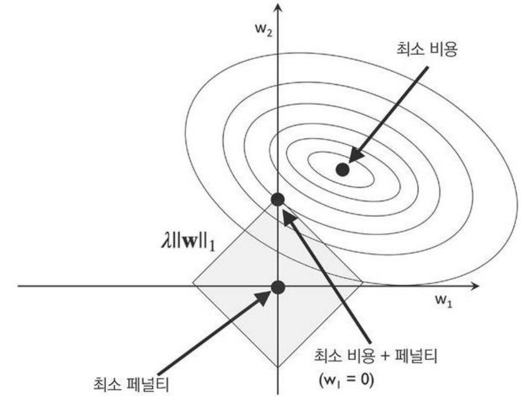
- 식 (5.34)의 가중치 감쇠 효과



(a) $\mathbf{sign}(\boldsymbol{\Theta}) = (1,1)^{\mathrm{T}}$인 경우 (b) $\mathbf{sign}(\boldsymbol{\Theta}) = (-1,1)^{\mathrm{T}}$인 경우

그림 5-22 *L1* **놈을 사용한 가중치 감쇠 기법의 효과**

최적화 해석

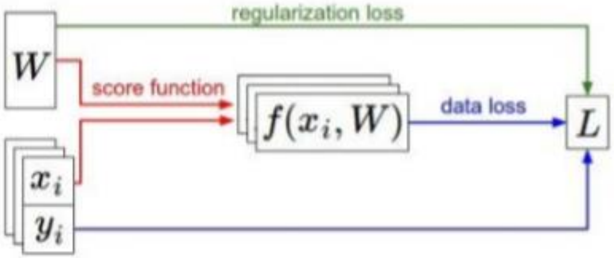


- $L1$ 놈의 희소성sparsity 효과 (0이 되는 매개변수가 많음)
  - 선형 회귀에 적용하면 특징 선택feature selection 효과

5.4.1 가중치 벌칙

규제

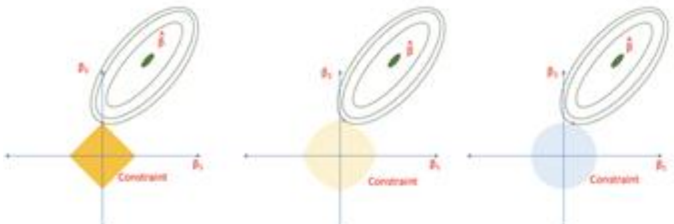
$$\underbrace{J_{regularized}(\boldsymbol{\Theta};\mathbb{X},\mathbb{Y})}_{\text{규제를 적용한 목적함수}} = \underbrace{J(\boldsymbol{\Theta};\mathbb{X},\mathbb{Y})}_{\text{목적함수}} + \lambda\,\underbrace{R(\boldsymbol{\Theta})}_{\text{규제 항}}$$



- 목적함수: 적용된 충분한 학습 모델로 훈련집합의 예측한 오차
- 규제: 학습 모델이 훈련집합의 예측을 너무 잘 수행하지 못하도록 방지

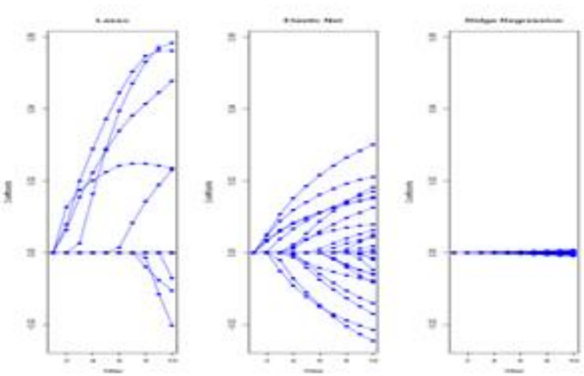
효과

- 가중치에 대한 선호도 표현
- 학습 모델을 단순화시킴으로 일반화 성능 향상 시킴
- 매끄럽게 하여 최적화 개선



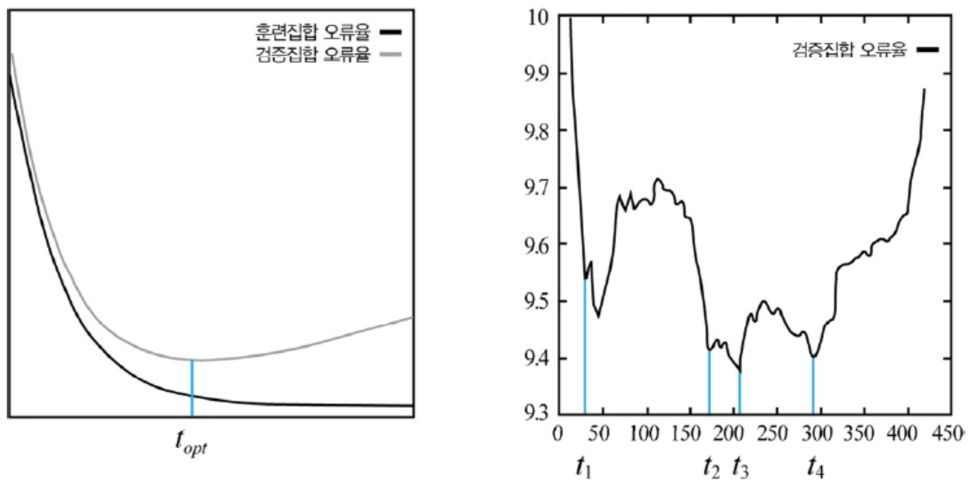
대표적인 예

- L2 규제: $R(W)=\sum_k\sum_l W_{k,l}^2$
- L1 규제: $R(W)=\sum_k\sum_l |W_{k,l}|$
- 엘라스틱 넷elastic net: $R(W)=\sum_k\sum_l \beta W_{k,l}^2 + |W_{k,l}|$(L1+L2)



5.4.2 조기 멈춤

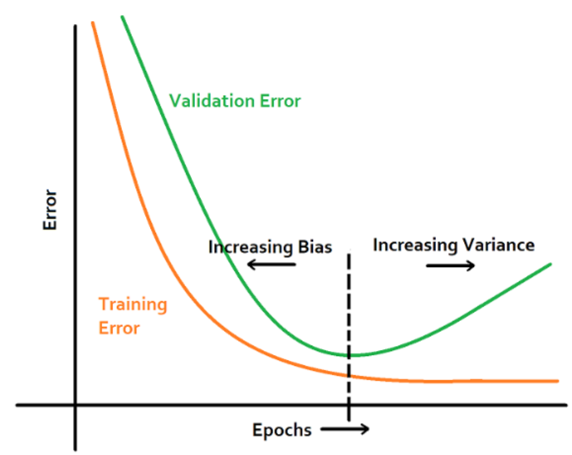
- 학습 시간에 따른 일반화 능력 [그림 5-23(a)]
  - 일정 시간 (t_{opt})이 지나면 과잉적합 현상이 나타남 → 일반화 능력 저하
  - 즉 훈련 데이터를 단순히 암기하기 시작



(a) 개념적인 도표 (b) 실제 데이터에 나타나는 지그재그 현상

그림 5-23 학습 시간에 따른 성능 추이

- 조기 멈춤early stopping이라는 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 현실은 [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
    - 참을성: 연속적으로 성능 향상이 없으면 멈추는 정도

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 $\hat{t}$

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

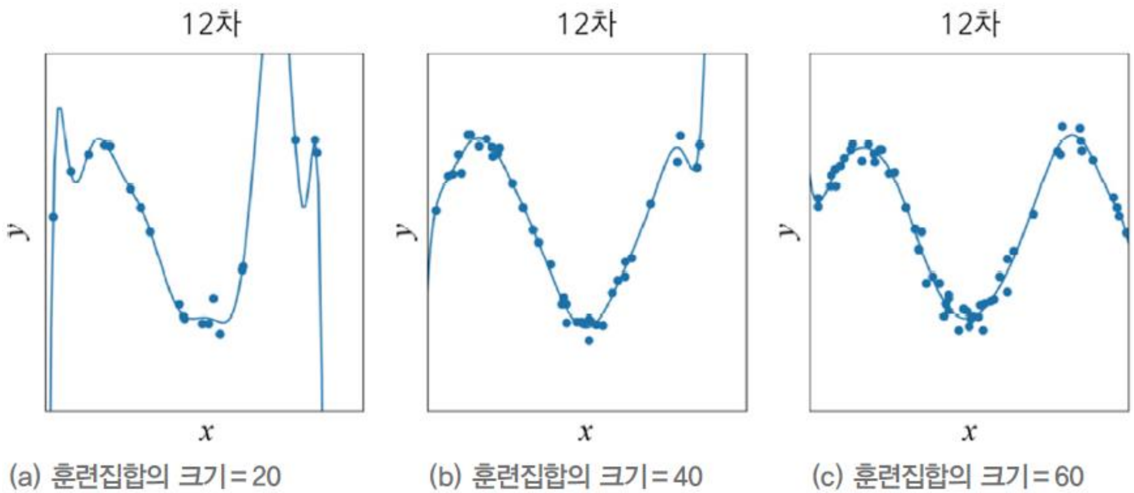


그림 1-17 데이터를 확대하여 일반화 능력을 향상함

  - 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 아핀affine 변환(이동translation, 회전rotation, 반전reflection)을 적용

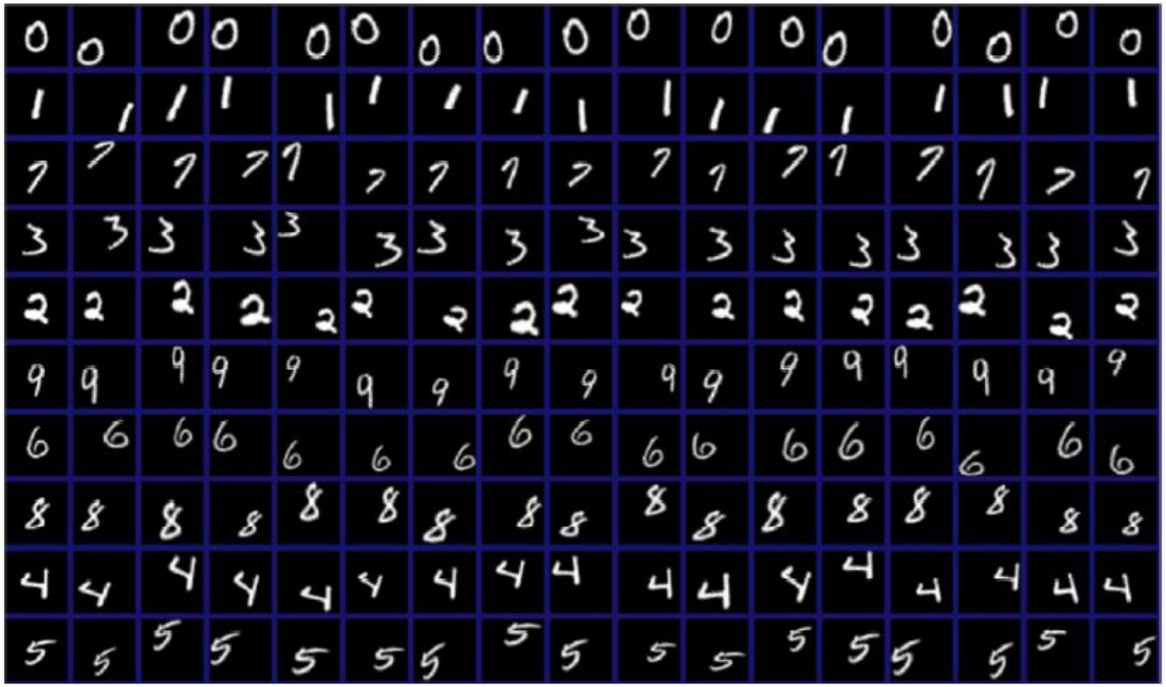


그림 5-24 필기 숫자 데이터의 다양한 변형[8]

- 한계
  - 수작업 변형
  - 모든 부류가 같은 변형 사용

5.4.3 데이터 확대

■ 예) 모핑morphing을 이용한 변형 [Hauberg2016]

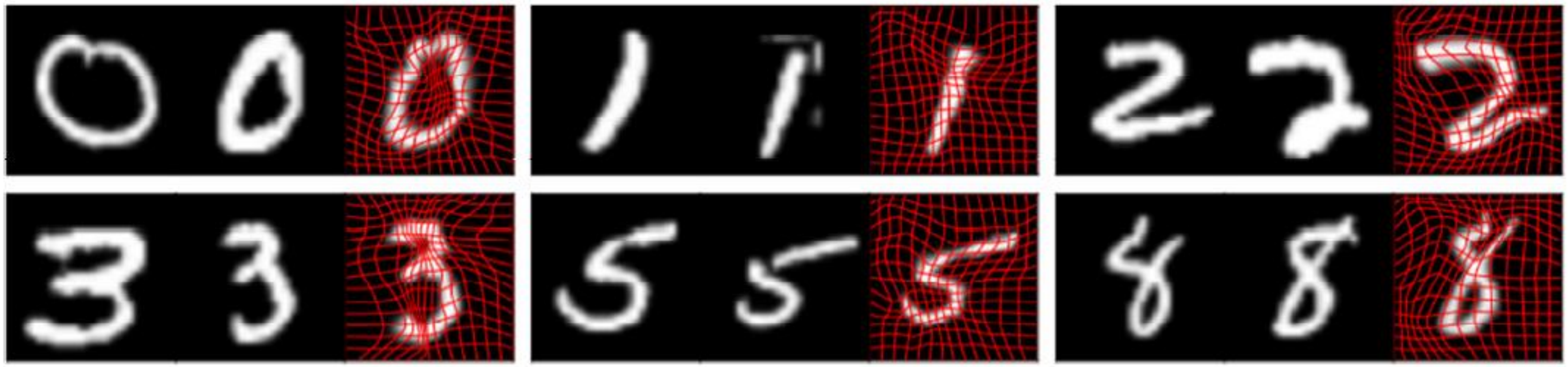
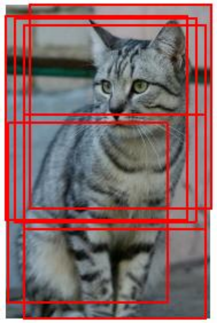


그림 5-25 비선형 변환 학습

- 비선형 변환으로서 아핀 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 변환color jitter으로 추가 확대
  - 예측 단계에서는 [그림 5-26]과 같이 5장 잘라내고 좌우 반전하여 10장을 만든 후, 앙상블 적용하여 정확도 향상

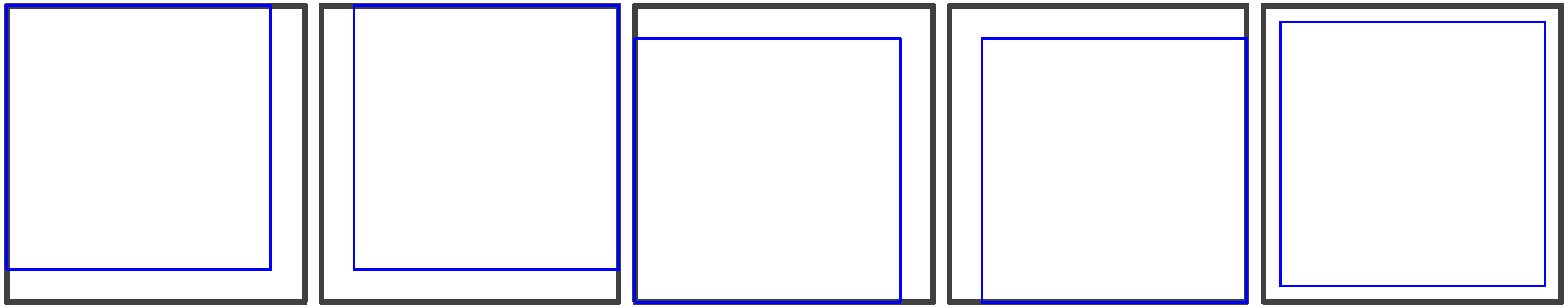


그림 5-26 예측 단계에서 영상 잘라내기

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