기계학습

5강 신경망(2)

장필훈 교수

학습목차

1 네트워크 훈련

2 오차역전파

3 정규화

01

네트워크훈련

1-1 네트워크훈련

- 매개변수를 정하기 위한 간단한 방법:

  제곱합 오류 최소화 $$E(\mathbf{w}) = \frac{1}{2}\sum_{\{n=1\}}^{N} \|y(\mathbf{x_n}, \mathbf{w}) - \mathbf{t_n}\|^2$$

- 회귀문제를 푸는 네트워크를 가정(앞서 나온것들):

  t가 $\mathbf{x}$에 종속인 평균을 가지는 가우시언을 따르면,

$$p(t|\mathbf{x}, \mathbf{w}) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1})$$

1-1 네트워크훈련

- (cont.)

  N개의 iid관측값$\mathbf{X} = \{\mathrm{x}_1, \dots, \mathrm{x}_\mathrm{N}\}$과 그에 해당하는

  표적값 $\mathbf{t} = \{\mathrm{t}_1, \dots, \mathrm{t}_\mathrm{n}\}$에 대해 가능도 함수를 구하고

  음의 로그: $$\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n, w) - t_n\}^2 - \frac{N}{2}\ln\beta + \frac{N}{2}\ln(2\pi)$$

- 회귀문제의 네트워크는 출력함수가 따로 필요 없다

  (항등함수를 씀. 확률적 해석이 필요하지 않기 때문)

1-1 네트워크훈련

- 뉴럴넷은 보통 오류함수를 최소화한다
  (음의 로그 최대화가 아니고)
  - 본질적인 차이는 없음
- 다음 식을 최소화한다.

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(\mathbf{x_n}, \mathbf{w}) - t_n\}^2$$

1-1 네트워크훈련

- β값은 $\mathbf{w}$를 찾은 뒤에 음의 로그 최소화로 찾음.
- 오류함수와 활성화함수 사이에는 짝이 있다.
  - Canonical link function에서 본것
  - 제곱합 오류함수 → (출력은) 항등함수
  - 이때 가지는 성질

$$\frac{\partial E}{\partial a_k} = y_k - t_k$$

1-1 네트워크훈련

- 이진분류(binary classification)문제
  - 다음 네트워크를 가정 $y = \sigma(a) = \dfrac{1}{1 + \exp(-a)}$
  - 출력값 자체를 확률로 볼 수 있다.
  - 출력값의 조건부 분포:

$$p(t|\mathbf{x}, \mathbf{w}) = y(\mathbf{x}, \mathbf{w})^t \{1 - y(\mathbf{x}, \mathbf{w})\}^{1-t}$$

1-1 네트워크훈련

- 조건부 분포에 음의로그(=오류함수)

$$E(\mathbf{w}) = -\sum_{\{n=1\}}^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}$$

=교차 엔트로피 오류함수
cross entropy

- 분류에서 cross entropy를 사용하면 제곱오류함수를 사용할 때보다 훈련과정이 더 빠르고, 일반화도 개선됨이 알려져 있다.

1-1 네트워크훈련

- 문제의 종류에 따른 출력과 오류 함수

	출력	오류함수
회귀	선형	제곱합오류함수
이진분류	로지스틱 시그모이드	교차엔트로피
다중클래스분류	소프트맥스	다중클래스 교차엔트로피

$$E(\mathbf{w}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_k(\mathbf{x_n}, \mathbf{w})$$

1-2 매개변수 최적화

- $E(\mathbf{w})$를 최소화 하는 $\mathbf{w}$를 찾는 문제
- $E(\mathbf{w})$가 $\mathbf{w}$에 대해 연속이고 미분가능하면,
  최솟값은 $\nabla E(\mathbf{w}) = 0$인 지점이다.
- 신경망의 경우 이 지점은 보통 여러개 존재한다.
- 비선형성이 해석적 해를 찾는 것을
  불가능에 가깝게 만든다.

1-2 매개변수 최적화

- 「연속적 비선형함수의 최적화」 문제
  - 보통 $\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \Delta\mathbf{w}^{(\tau)}$를 이용하고,

    $\Delta\mathbf{w}^{(\tau)}$는 $\Delta E(\mathbf{w})$에 의존한다.
  - 구체적인 예:

    지역적 이차근사문제

1-3 지역적 이차근사

- 어떤 점 $\hat{\mathbf{w}}$에 대한 $E(\mathbf{w})$의 테일러 전개

$$E(\mathbf{w}) \simeq E(\hat{\mathbf{w}}) + (\mathbf{w} - \hat{\mathbf{w}})^T \mathbf{b} + \frac{1}{2}(\mathbf{w} - \hat{\mathbf{w}})^T \mathbf{H}(\mathbf{w} - \hat{\mathbf{w}})$$

$\mathbf{H}$는 헤시안 $= \nabla\nabla E$

$$\mathbf{b} \equiv \nabla \mathrm{E}\Big|_{\mathrm{w}=\hat{\mathrm{w}}} \qquad (\mathbf{H})_{ij} \equiv \frac{\partial E}{\partial w_i \partial w_j}\Big|_{\mathbf{w}=\hat{\mathbf{w}}}$$

$$\therefore \nabla E \simeq \mathbf{b} + \mathbf{H}(\mathbf{w} - \hat{\mathbf{w}})$$

기울기의 지역적 근사치

1-4 경사하강 최적화

- 기울기정보를 사용하는 가장 단순한 방법:

  gradient descent

  $$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta \nabla E\left(\mathbf{w}^{(\tau)}\right), \quad \text{학습률 } \eta > 0$$

- 오류함수는 훈련집합에 따라 계산되므로

  ∇E를 계산하려면 전체 훈련집합을 계산해야 함

  : 배치방식

1-4 경사하강 최적화

- 단순한 형태로는 좋은 결과를 얻지 못하고,
  conjugate gradient등 더 좋은 방법이 존재한다.
- **online version**
  - 데이터가 너무 클 때
  - stochastic gradient descent
    - 한번에 데이터 하나에 대해 가중치를 업데이트

1-4 경사하강 최적화

- 온라인배치방법의 장점
  1. 데이터상의 중복처리가 효율적이다.
     중복된 데이터가 많을 경우 배치방법은 반복계산이 많아진다.
  2. 지역적최솟값에서 탈출하기가 쉽다
     오류함수 임계점: 개별포인트 ≠ 전체 데이터

02

오차역전파

2-1 오차역전파

- error back-propagation
- 두 단계로 이루어진다
  1. 오류함수 미분의 계산
  2. 계산 결과를 바탕으로 가중치($\mathbf{w}$) 조절
- 미분의 계산(예시)

  ① 단순 선형모델 가정 $y_k = \sum_i w_{ki} x_i$

2-1 오차역전파

- 미분의 계산(예시)

특정 n에 대한 오류함수 $E_n = \frac{1}{2}\sum_k (y_{nk} - t_{nk})^2$

where $y_{nk} = y_k(\mathbf{x}_n, \mathbf{w})$

이 때, w_{ji}에 대한 ∇E_n

$$\frac{\partial E_n}{\partial w_{ji}} = (y_{nj} - t_{nj})x_{ni}$$

2-1 오차역전파

- 미분의 계산(예시)

② 일반적 피드포워드 네트워크의 경우

$$z_j = h(a_j), a_j = \sum_i w_{ji} z_i$$

h: 비선형 활성함수

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j} \cdot \frac{\partial a_j}{\partial w_{ji}} \quad , \qquad \frac{\partial a_j}{\partial w_{ji}} = \frac{\partial}{\partial w_{ji}} \sum w_{ij} z_i = z_i$$

2-1 오차역전파

- 미분의 계산(예시)

해당 유닛의 출력값

$$\therefore \frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j} z_i$$

해당유닛의 입력값에의한 값

→선형모델의 유닛과 같다.

- 이것을 층(layer)마다 반복해서 모든 유닛의 w계산.

2-1 오차역전파

- 미분의 계산(예시)
  - 은닉유닛을 계산하려면 편미분을 한번 더 하면 된다.

$$\frac{\partial E_n}{\partial a_j} = \sum_k \frac{\partial E_n}{\partial a_k} \cdot \frac{\partial a_k}{\partial a_j}$$

$$\delta_j = h'(a_j) \sum_k w_{kj} \delta_k \quad \left(\delta_j = \frac{\partial E_n}{\partial a_j} \right)$$

  - 연쇄성에 주목.(k가 앞단, j가 뒷단)

2-1 오차역전파

- 계산 과정

1. 입력을 주고 모든 은닉유닛과 출력유닛의 값을 계산
2. 모든 출력유닛의 $\frac{\partial E_k}{\partial a_k}$ 계산
3. 에러를 역으로 거슬러 올라가서 모든 은닉유닛의 $\frac{\partial E_k}{\partial a_j}$ 계산
4. 미분계산

2-1 오차역전파

- 예시2
  - 활성함수

    $$h(a) \equiv \tanh(a) = \frac{e^{a}-e^{-a}}{e^{a}+e^{-a}}, \qquad \tanh'(a) = 1 - \tanh^{2}(a)$$

  - 오류함수

    $$E_n = \sum_{k=1}^{K}(y_k - t_k)^2$$

2-1 오차역전파

- 예시2(cont.)

$$a_j = \sum_{i=0}^{D} w_{ji}^{(1)} x_i, \qquad z_j = \tanh(a_j), \qquad y_k = \sum_{j}^{M} w_{kj}^{(2)} z_j$$

출력유닛이 canonical link function이므로,

각 출력유닛에서 $\frac{\partial E_n}{\partial a_k} = y_k - t_k$

역전파 $\frac{\partial E_n}{\partial a_j} = (1 - z_j^2) \sum_{k=1}^{K} w_{kj} \cdot \frac{\partial E_n}{\partial a_k}$

2-1 오차역전파

- 예시2(cont.)
  - 첫번째 계층에서, $\frac{\partial E_n}{\partial w_{ji}^{(1)}} = \frac{\partial E_n}{\partial a_j} x_i$
  - 두번째 계층에서, $\frac{\partial E_n}{\partial w_{kj}^{(2)}} = \frac{\partial E_n}{\partial a_k} z_j$

03

정규화

3-1 정규화

- 뉴럴넷에서 입력과 출력의 차원은 데이터에 의해 정해짐
- hidden unit의 수나 모양은 마음대로 정할 수 있음.
    - 모델비교를 통해 결국 결정하는 때가 많음
- 앞서 선형회귀문제에서는 차수를 적당히 충분히 잡고 regularization term을 넣는 방법을 씀
- 뉴럴넷은 dropout등을 씀

3-1 정규화

- 제곱정규화항(가중치 감쇠) 고찰

$$\tilde{E}(w) = E(w) + \frac{\lambda}{2} w^T w$$

문제점: 가중치(w)의 값 크기 자체에 의존한다.

3-1 정규화

- 제곱정규화항의 문제점(cont.)
  - 입력값을 선형변환 후
    구조가 같은 다른 넷의 훈련데이터로 사용한다고 가정.

$$x_i \rightarrow \tilde{x}_i = ax_i + b$$

  - 은닉유닛들의 w를 다음과 같이 조정하면,

$$w_{ji} \rightarrow \widetilde{w}_{ji} = \frac{1}{a} w_{ji}, \qquad w_{j0} \rightarrow \widetilde{w}_{j0} = w_{j0} - \frac{b}{a} \sum_i w_{ji}$$

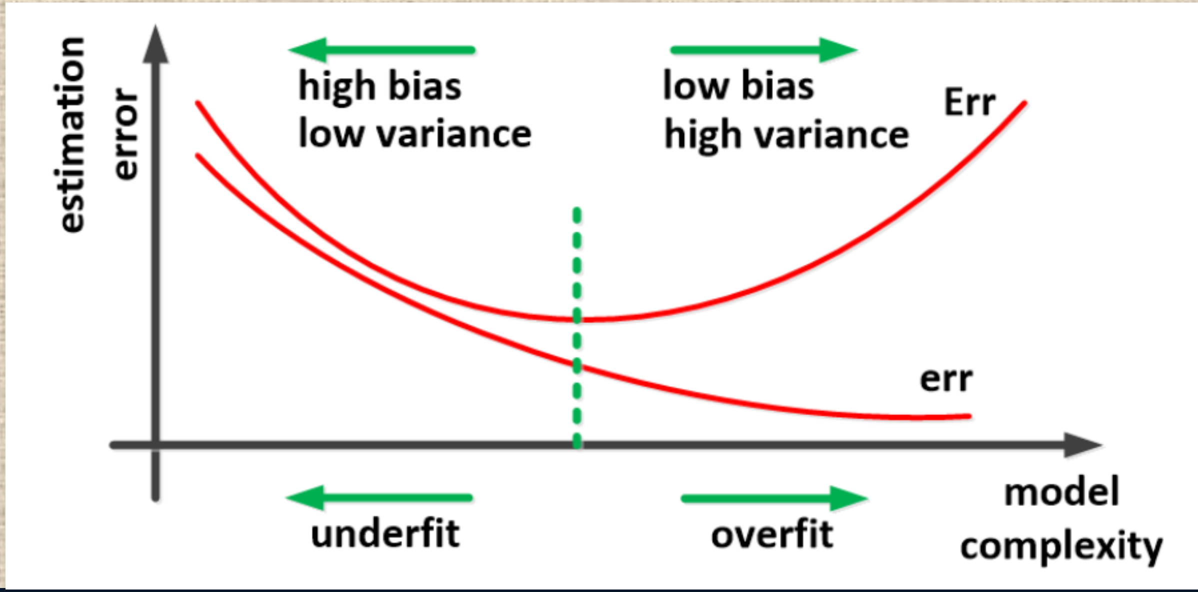
3-1 정규화

- 조정하면, 두 네트워크(선형변환 이전과 이후 각각의 값으로 훈련시킨 두 네트워크)의 사상이 일치한다.
  projection

- 제곱정규항은 그렇지 않음. 그래서 다른 형태를 쓴다.

3-1 정규화

- **조기종료**
  - 네트워크 복잡도를 조절하기 위한 다른 방법
- 훈련집합 오류와 검증집합 오류의 오차를 본다.



Ghojogh, Benyamin, and Mark Crowley. "The theory behind overfitting, cross validation, regularization, bagging, and boosting: tutorial." arXiv preprint arXiv:1905.12787 (2019).APA , Fig 6(a)

3-1 정규화

- **불변성**
  - 이동불변성, 크기불변성 등
  - 학습 가능하다
    1. 데이터 자체변환
    2. 입력변환시 출력이 변하는 것에 불이익을 준다
    3. 불변성을 사전처리과정에 추가한다.(ex.특징추출)
    4. 뉴럴넷 자체구조에 불변성 포함(ex. CNN)

3-1 정규화

- 불변성 학습방법들의 특징
  1. 계산량이 많다.
  2. 정규화항만 추가하면 된다.
     - 랜덤노이즈를 더하는 것과 비슷한 것으로 알려짐
  3. 훈련데이터에 없는 변환도 학습된다.

3-1 정규화

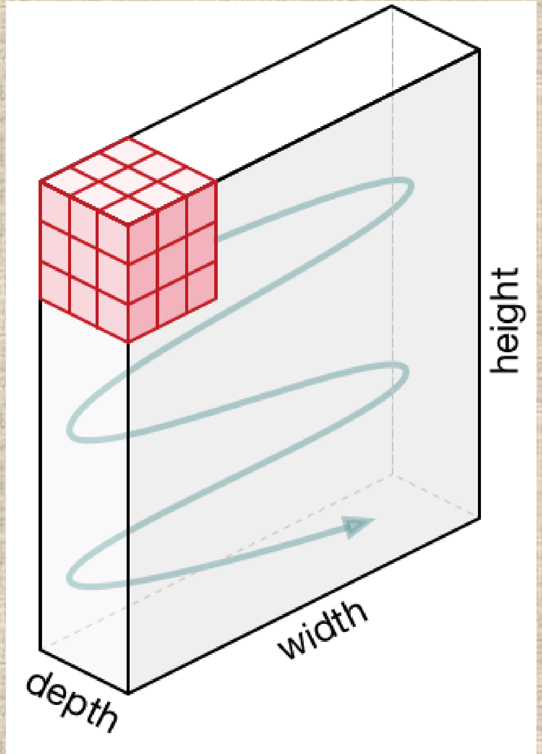
- **CNN** (Convolutional Neural Net)
- 숫자인식의 경우 모든 입력값이 연결된 net 사용가능
  - 데이터가 충분해야 한다.
    - 모든 변환을 포함하고 있을 정도로
    - 불가능에 가깝다
- FCN보다는 이미지의 지역적 특징을 이용해야 한다

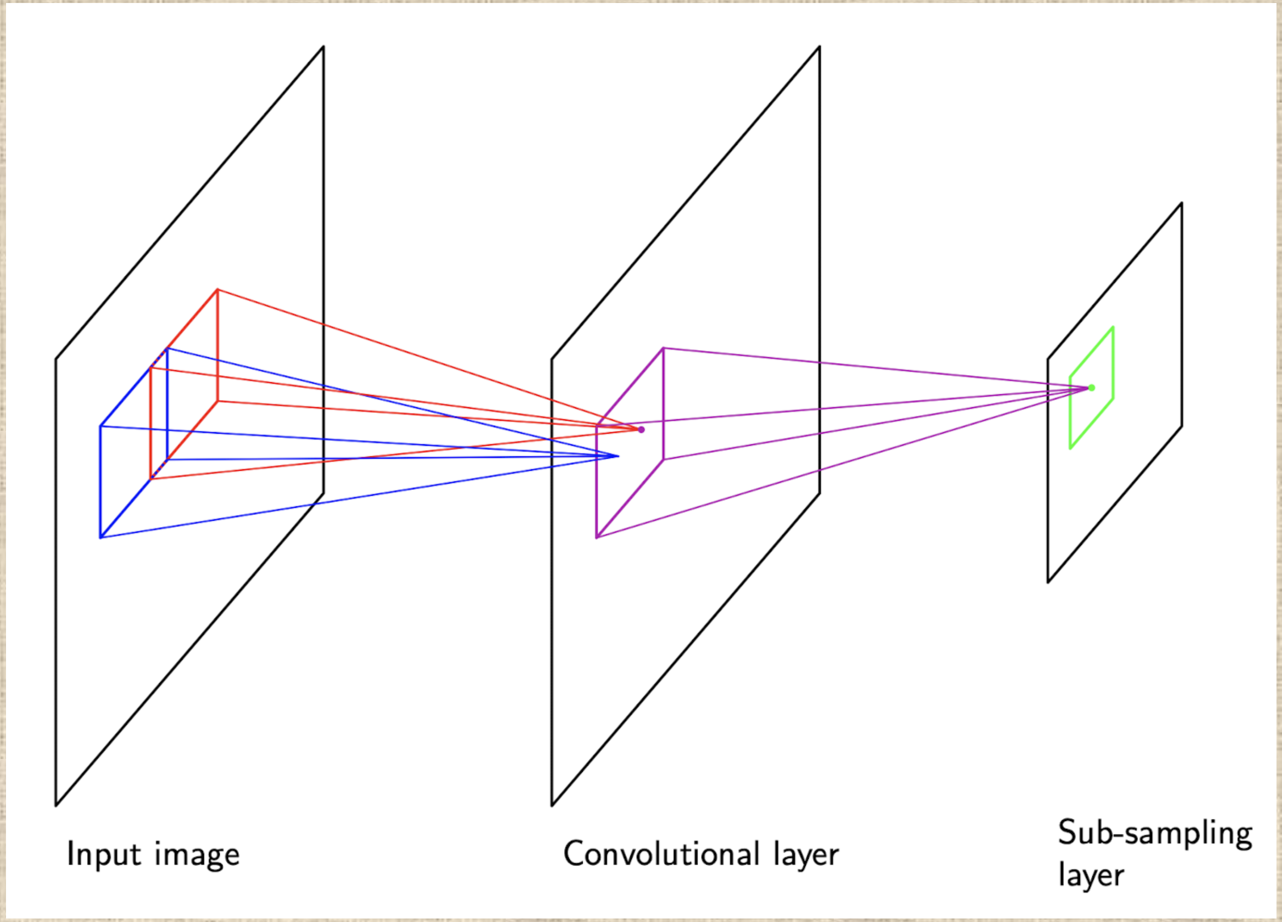
3-1 정규화

- CNN(Cont.)
  - 지역적 특성 : 가까이 있는 픽셀들은 멀리 있는 픽셀들에 비해 밀접하게 연관되어 있다.
  - CNN은 이런 특징을 반영할 수 있다.

3-1 정규화

- CNN의 구조





Svensén, Markus, and Christopher M. Bishop. "Pattern recognition and machine learning." (2007). Fig. 5.17

Sumit Saha, 「A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way」,
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

3-1 정규화

- CNN의 구조(Cont.)
  - ‘필터’ 하나를 feature map하나로 생각할 수 있다.
  - 각 계층은 이전층에서 추출된 부분적 특성을 추상화, 구조화한다. (불변성을 갖추게 됨)
  - 가중치를 공유하는 효과가 있어서 계산상 유리함.

3-2 혼합밀도네트워크

- 지금까지는 조건부 분포 $p(\mathbf{t}|\mathbf{x})$를 모델링 할 때, 가우시안을 가정하고 식을 전개
- 가우시안이 아니면? 근사.
  - 혼합밀도 네트워크- 아래는 이분산성 모델의 예시

$$p(\mathbf{t}|\mathbf{x}) = \sum_{k=1}^{K} \pi_k(\mathbf{x})\mathcal{N}(\mathbf{t}|\boldsymbol{\mu}_k(x), \mathbf{I}\sigma_k^2(\mathbf{x}))$$

혼합계수 π_x

3-2 혼합밀도네트워크

- 가우시안 대신 다른 분포를 성분으로 사용 가능
- 오류함수의 미분을 계산할수만 있다면, 매개변수 w 를 구할 수 있다.

3-3 베이지안 뉴럴 네트워크

- 선형회귀에서 베이지안 최대가능도법을 이용, w계산
- 다계층 네트워크의 경우 매개변수값에 대한 비선형성으로 정확한 베이지안 해를 찾는 것이 불가능하다.
  - 근사해서 구한다

다음시간

6강

- 커널방법론
- SVM