

비타민 3주차 정규세션

오버피팅과 규제


BITAmin 5조
최대상 조은정



CONTENTS —

01 오버피팅

02 다양한 규제 기법

- 가중치 감소
 - 드롭아웃
 - 배치정규화
 - 조기 종료
- 



01

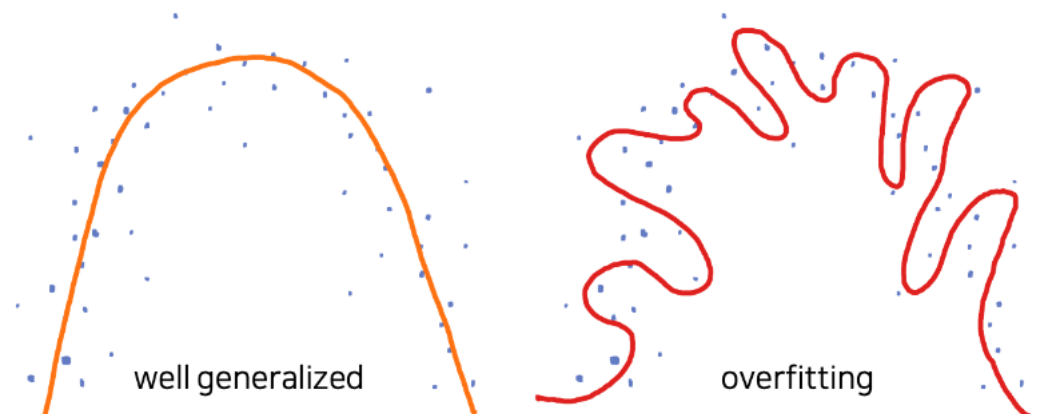
오버피팅

Overfitting

- 학습 오차가 일반화 오차에 비해서 현격하게 낮아지는 현상

-> 모델이 학습 데이터의 불필요한 편향이나 노이즈까지 학습함으로써 모델의 일반화 성능이 떨어지는 상황

따라서 오버피팅을 최소화 하는 것이 모델링에서 가장 중요한 이슈!






규제

Regularization

오버피팅 01

- 오버피팅을 낮추고 모델이 학습 데이터로부터 적절한 특징들을 학습하여 일반화 오차를 낮추는 기법
 - 모델 복잡도를 줄이기 위해 제약을 두는 방법
- 

— 02

다양한 규제 기법

다양한 규제 기법

다양한 규제 기법 02

규제 Regularization

데이터를 통한 규제

데이터 증강

손실 함수를 통한
규제

가중치 감쇠

신경망 계층을 통한
규제

드롭아웃

배치정규화

계층정규화

학습/추론 방식을
통한 규제

학습 조기 종료

배깅 & 앙상블

가중치 감쇠

Weight decay

다양한 규제 기법 02

- 학습 과정에서 큰 가중치에 대해 그에 상응하는 페널티를 부과하는 방식
이유: 오버피팅은 가중치 매개변수의 값이 커서 발생하는 경우가 많음
- 손실 함수를 수정하는 방법을 통해 적용

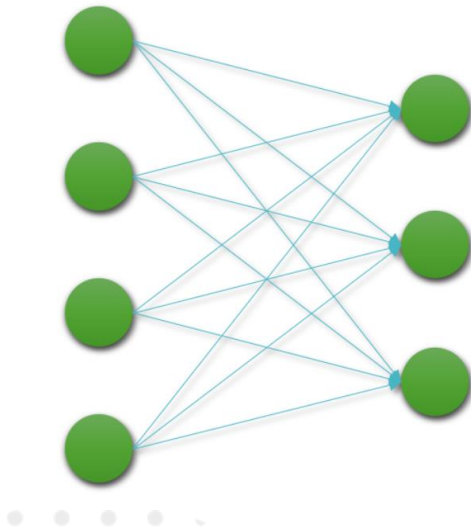
$$\begin{aligned}\tilde{\mathcal{L}}(\theta) &= \mathcal{L}(\theta) + \alpha \|W\|_2^2 \\ &= \mathcal{L}(\theta) + \alpha W^\top \cdot W, \\ \text{where } \theta &= \{W, b\}.\end{aligned}$$

- 신경망 학습 목적: 손실 함수 값 줄이기
- 가중치 제곱 노름(L2 norm)을 손실 함수에 더함
-> 가중치 증가 억제 가능

가중치 감쇠

Weight decay

- 손실 함수를 수정하는 방법을 통해 적용



$$\begin{aligned}\tilde{\mathcal{L}}(\theta) &= \mathcal{L}(\theta) + \alpha \|W\|_2^2 \\ &= \mathcal{L}(\theta) + \alpha W^\top \cdot W, \\ \text{where } \theta &= \{W, b\}.\end{aligned}$$

- 가중치 파라미터에서 W 의 각 요소:
선형계층에서 입출력 노드 사이의 관계
- 요소 값의 크기가 클수록 강한 관계임을 의미
- W 에 L2 노름을 취해서 관계 약화
-> 노드 사이의 관계의 강도 제한
-> 각 계층의 출력 노드가 다수의 입력 노드로부터 과도하게 학습하는 것을 제한할 수 있음

가중치 감쇠

Weight decay

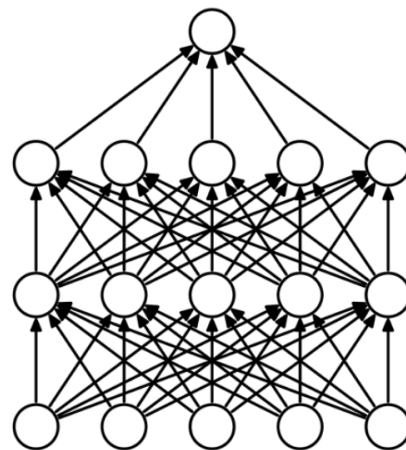
다양한 규제 기법 02

$$\tilde{\mathcal{L}}(\theta) = \underbrace{\mathcal{L}(\theta)}_{(1)} + \underbrace{\alpha \|W\|_2^2}_{(2)}$$

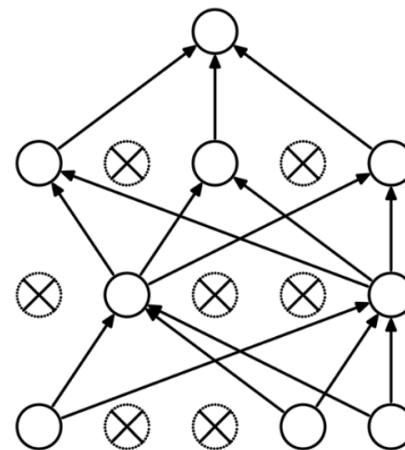
- (1)이 커지면 (2)가 작아지고 (2)가 커지면 (1)이 작아짐
-> (2)의 스케일을 설정하는 하이퍼 파라미터 α 의 역할
중요
- 스케일 설정 값이 0에 가까워질수록 (1)을 최소화하기
쉽고, 설정 값이 커질수록 가중치 파라미터가 W(기존
손실 함수의 가중치)에 가까워지기 어려움
-> 해당 하이퍼 파라미터에 따라 모델 성능 결정
-> 학습을 방해하여 오버피팅 최소화

드롭아웃 Dropout

- 임의의 노드를 일정 확률로 드롭해서 학습에 참여하지 않도록 하는 방법
 - 훈련을 반복할 때마다 신경망의 뉴런 출력을 랜덤하게 0으로 만드는 작업
 - 매훈련마다 조금씩 다른 뉴런 연결 방식(토폴로지)
- > 신경망이 각 입력 샘플을 암기하려는 기회 감소-> 오버피팅 방지



(a) Standard Neural Net

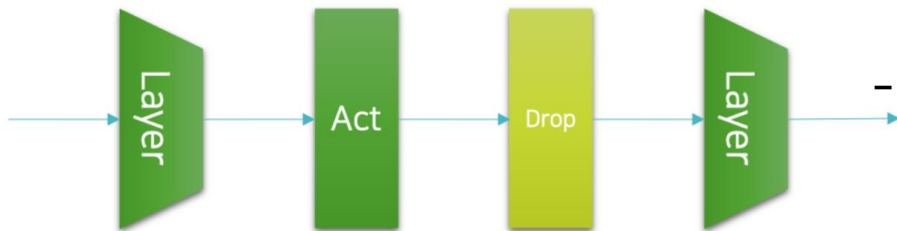


(b) After applying dropout.

드롭아웃 Dropout

다양한 규제 기법 02

- 구현 방식



- 비선형 활성화 함수와 선형 혹은 여러 계층의 컨볼루션 모듈 사이에 nn.Dropout 모듈을 넣어 구현
- 인자: 어떤 입력이 0이 될지에 대한 확률 p 지정
(p 는 보통 0.1~0.5 사이에서 0.1 단위로 튜닝)
- 훈련할 때만 사용하고 평가와 예측 시에는 사용하지 않음

- 한계

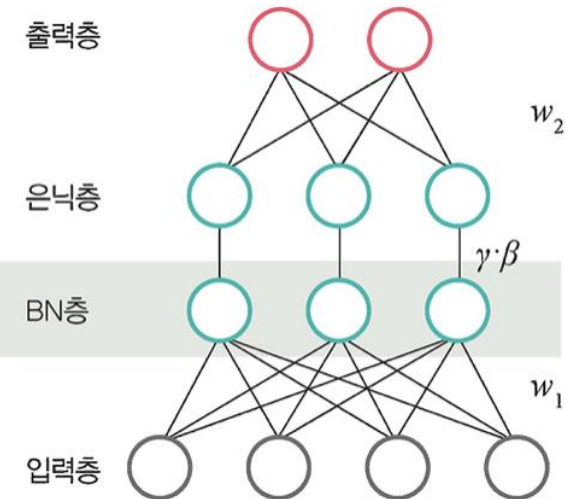
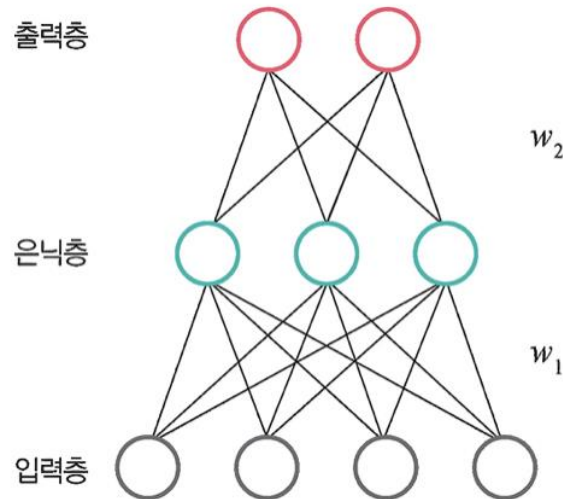
- 손실 값의 수렴 속도 저하 (훈련 시간 증가)
- 학습 오차 증가

배치정규화

Batch normalization

다양한 규제 기법 02

- 기존 규제 방식의 문제: 학습을 방해하는 형태로 작용하여 학습 및 수렴 속도 저하
- 배치정규화
 - 학습 속도를 향상시킬 뿐만 아니라 일반화 성능까지 개선할 수 있는 방법
 - 현재 가장 많이 사용하는 규제 방법



배치정규화

Batch normalization

다양한 규제 기법 02

$$\text{batch_norm}(x) = \gamma \frac{(x - \mu)}{\sqrt{\sigma^2 + \epsilon}} + \beta$$

$$\mu = x.\text{mean}(\text{dim} = 0)$$

$$\sigma = x.\text{std}(\text{dim} = 0)$$

where $x \in \mathbb{R}^{N \times n}$.

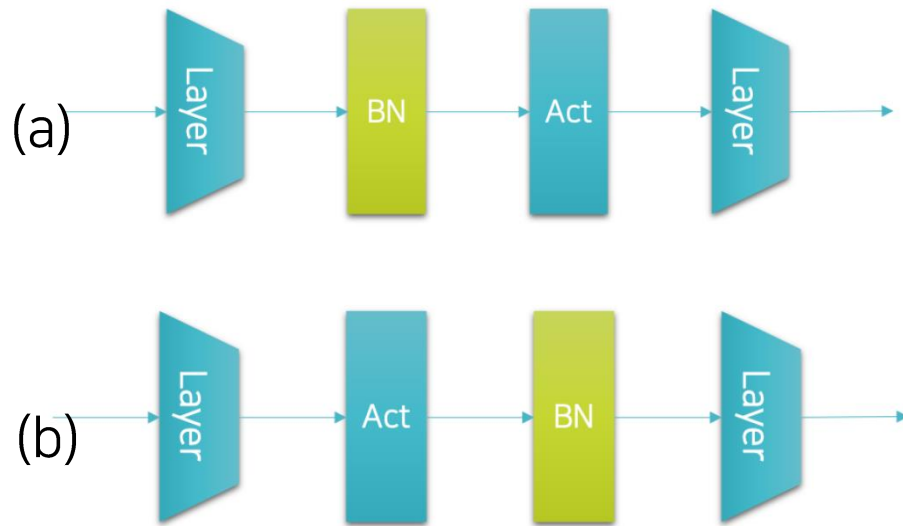
- 공변량 변화 문제 해결
 - 공변량 변화: 네트워크의 각 층마다 활성화 함수가 적용되면서 입력값의 분포가 계속 바뀌는 현상
 - 해결 방법: 분산된 분포를 정규분포로 만들
 - 미니배치를 가우시안 분포로 정규표준분포화
 - > 스케일 파라미터 γ 와 이동 파라미터 β 적용 (γ, β : 가중치 파라미터)

배치정규화

Batch normalization

다양한 규제 기법 02

- 구현



- 단점

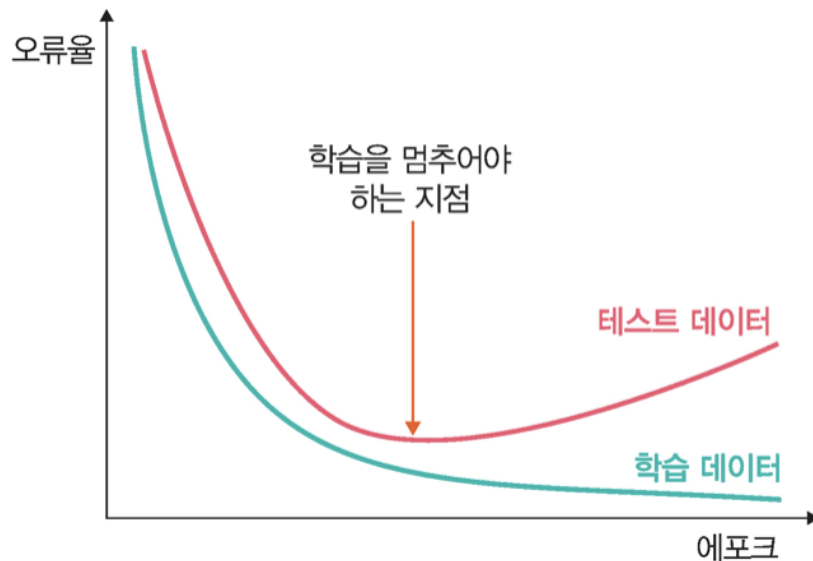
- 신경망 내부에 배치정규화 계층 삽입
 - 주로 선형 계층과 활성화 함수 사이에 삽입(a)
 - 다만 드롭아웃의 삽입 위치와 동일하게 삽입하는 (b)의 방식도 가능
-
- 배치 크기가 작을 경우 정규화 값이 기존 값과 다른 방향으로 훈련될 수 있음
(ex. 분산이 0이면 정규화 자체가 안 될 수 있음)
 - RNN의 경우 네트워크 계층별로 미니 정규화를 적용해야 함-> 모델이 더 복잡해지면서 비효율적일 수 있음

조기종료

Early stopping

다양한 규제 기법 02

- 뉴럴 네트워크가 오버피팅을 회피하는 규제 기법
- 매 에포크마다 검증 데이터에 대한 오차를 측정해서 모델의 종료 시점을 제어함



- 오버피팅 발생 전
 - 학습에 대한 오차, 검증에 대한 오차 모두 감소
 - 오버피팅 발생 후
 - 훈련 데이터셋에 대한 오차: 감소
 - 검증 데이터셋에 대한 오차: 증가
- > 따라서 검증 데이터셋에 대한 오차가 증가하는 시점에서 학습을 멈추도록 조정

- 한계

- 학습 종료 시점만 결정할 뿐 모델 성능을 보장하지 않음

비타민 3주차 정규세션

**THANK
YOU**