06. 배치 정규화

<배치정규화>

배치 정규화 :

- 하이퍼 파라미터 탐색을 쉽게 하고 신경망과 하이퍼파라마니터의 상관관계를 줄여준다
- 더 많은 하리퍼파라미터가 더 잘 작동하게 한다
- 깊은 심층신경망도 잘 작동하게 한다

입력변수를 정규화하면 학습이 빨라짐진다

ex) w[3],b[3]를 학습시키려면 a[2]의 평균과 분산을 정규화하는 것이 효율적! (a[2]가 input이기 때문)

⇨ 배치정규화가 하는 일

⇨ z[2]를 정규화하는 것

활성함수 이전의 값인 z[2] 를 해야하는지, 활성화 함수 이후 값인 a[2]를 정규화해야하는지 논쟁

★ **z**[2]가 더 자주 쓰인다 (디폴트)

<배치 정규화 구현하기>

- $\mu = \frac{1}{m}\sum_i z^{(i)}$
- $\sigma^2 = \frac{1}{m}\sum_i (z^{(i)} - \mu)^2$
- $z^{(i)}_{norm} = \frac{z^{(i)} - \mu}{\sqrt{\sigma^2 + \epsilon}}$
- $\tilde{z}^{(i)} = \gamma z^{(i)}_{norm} + \beta$

- 신경망에서 사잇값들이 주어졌다고 할 때 은닉유닛의 값 = z^[l](i) (i는 1부터m까지)

[l]생략(표기상 편의)

평균 이렇게 계산, 분산을 이렇게 계산.

각 z(i)를 정규화하여 z(i)_norm 을 얻는다! (평균을 빼고 분산으로 나누고, 분모가 0이 되지 않도록 분모에 앱실론 추가)

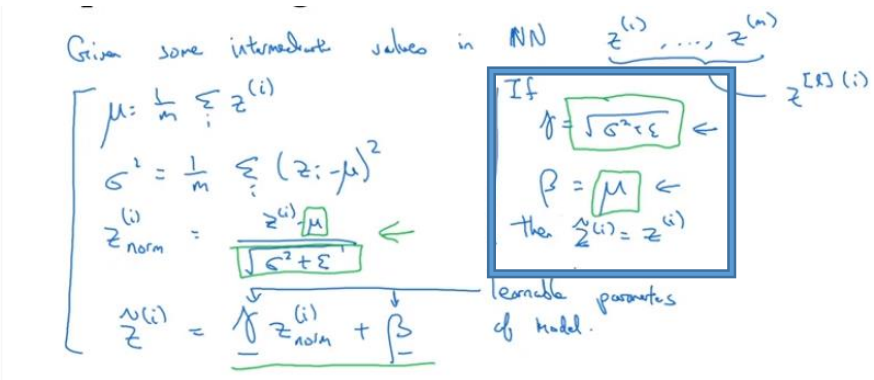
⇨ 정규화 거쳐서 표준편차 1 되도록 만든다

but 은닉 유닛이 항상 평균 0 표준편차 1을 갖는 것이 좋지만은 않다

대신 z~(i) = gamma*z^(i)_norm+beta

gamma,beta: 모델에서 학습시킬 수 있는 변수 (업데이트 해야한다)

⇨ z~의 평균을 원하는 대로 설정할 수 있다



값 외 다른 값 설정하면 은닉 유닛의 값들이 서로 다른 평균이나 분산 값을 만들게 할 수 있다

z(i) 대신 z~(i)를 쓴다

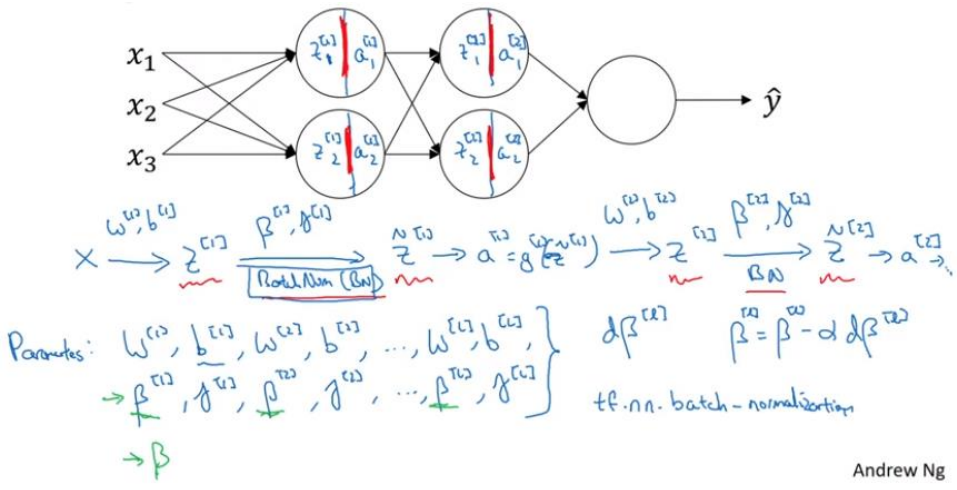
★ 입력층에만 정규화 하는 것이 아니라 깊이 있는 은닉층에도 정규화를 사용하여 은닉 유닛 z의 평균과 분산 정규화한다!

입력층 vs 은닉유닛

⇨ 은닉 유닛을 정규화할때에는 평균과 분산이 0,1,로 고정되기를 원치 않는다

ex) 시그모이드의 비선형성을 살릴 수 있도록 평균이 0이 아닌 다른 값을 가지게 하는 것이 좋다

<배치 정규화 적용시키기> 심층 신경망에 적용



- 은닉유닛은 2단계를 거친다 => z를 우선 계산하고 활성화함수로 a를 계산한다
- z[1]으로 BN으로 beta1, gamma1의 영향을 받아 새로 정규화된 z[1]얻고, a(1)=g[1](z[1])

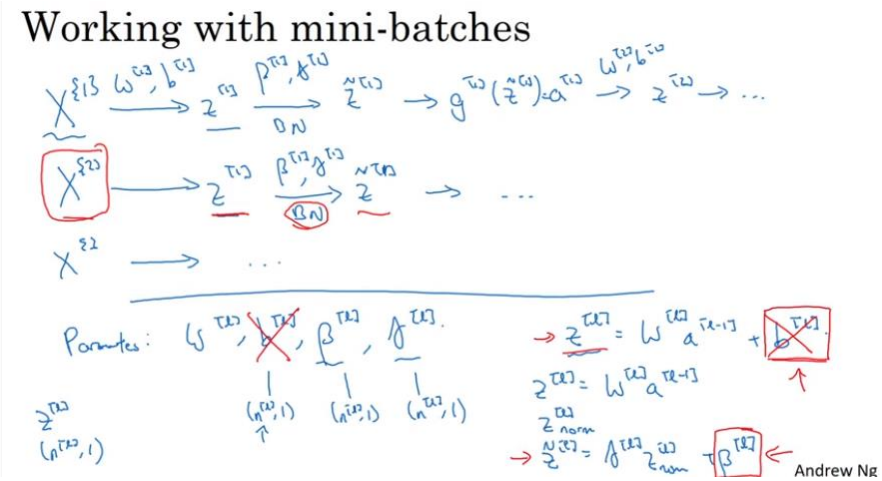
정규화된 z~를 이용한다!

주의) beta는 모멘텀이나 RMSProp 에 쓰이는 하이퍼 파라미터랑 다르다

beta=beta-learningrate*dbeta[l]

- 경사하강법, adam, rmsprop, 모멘텀 중에 선택하여 사용

(딥러닝 프로그래밍 프레임워크를 사용하면 ex) tf.nn.batch_normalization 같이 한줄로 구현 가능)



1. 실제로는 첫 미니배치에 대해 z[1] 계산하고 z[1]의 평균, 분산 계산 후

배치 정규화 진행 (beta[1],gamma[1]로 조정)

z[1] 구하고 활성화 함수에 넣어서 a[1]구하고

z[2]구하고.. 반복하여 첫번째 미니배치에 대한 경사하강법 수행

2. 두번째 미니배치에 대해서는 <u>두번째 미니배치만을</u> 이용해서 평균과 분산 계산

z[l]=w[l]*a[l-1]+b[l]

3. 미니배치를 보고 z[l]이 평균 0 분산 1 가지도록 정규화하고 beta랑 gamma로 조정

상수를 더해줘도 평균을 빼주면서 사라지기 때문에 영향을 끼치지 않는다

- ⇨ **<u>b[l]을 쓰지 않아도 된다!</u>**

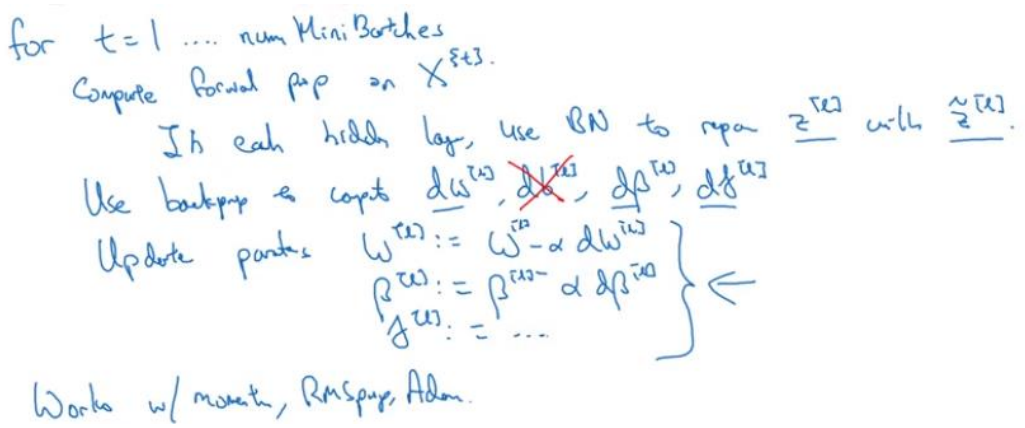
z[l]=w[l]*a[l-1] , z[l]norm, z~=gamma[l]*z[l]_norm+beta[l]

(beta[l]은 z~[l]의 평균을 조정하기 위해 쓰인다)

- z[l]의 dim : (n[l],1) , dim(b[l])=(n[l],1) (n은 은닉유닛 수)
- beta[l],gamma[l] 의 차원도 (n[l],1)

(각 은닉 유닛 값 조정하기 위해 쓰이는 값이라서)

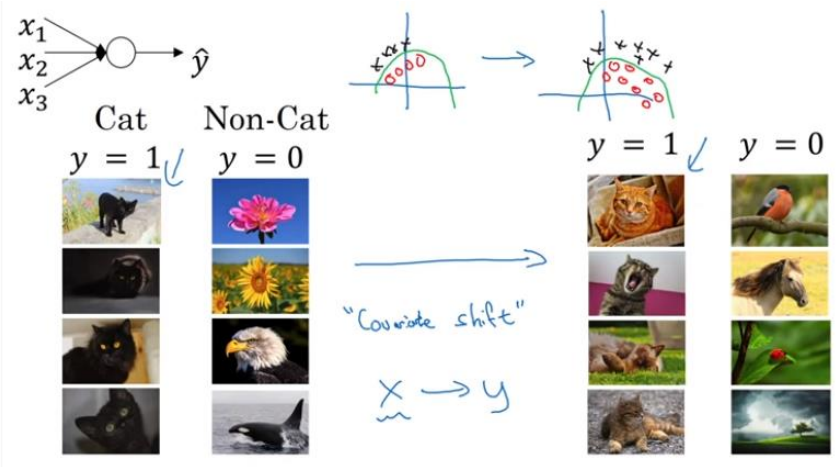
for t=1...number of minibatches



1. 정방향 전파 on X{t}
2. each hidden layer에서 BN을 사용하여 z[l]을 z~[l]을 이용한다
3. 역방향 전파를 이용하여 dw, ~~db~~, dbeta, dgama 계산
4. 파라미터 업데이트
5. 경사하강법 이용 (or 모멘텀, rmsprop, Adam)

배치정규화가 잘 작동하는 이유는 무엇일까요?

< cat detection task >



- 검정 고양이의 이미지만을 이용하여 학습시켰을 때
- 다른 색의 고양이가 정답일 경우 좋은 성능X

정답과 오답의 분포 오른쪽의 그래프이면 왼쪽 데이터로 학습시킨 모델이 오른쪽 데이터에서 좋은 성능 X

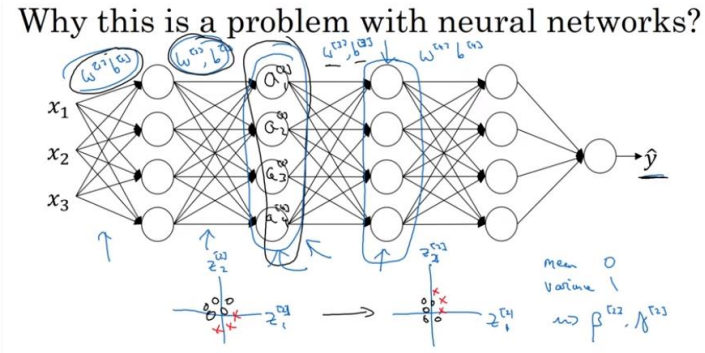
공변량 변화 : X,Y의 대응을 학습시킬 때 x의 분포가 바뀐다면 학습 알고리즘을 다시 학습해야한다

(관측함수가 바뀌지 않더라도)

< 배치정규화의 효과 >

1. 공변량 변화 문제 해결

< y hat으로 매핑할 때 >



- 은닉층 값들이 계속 바뀌므로 공변량 변화 문제를 계속 겪는다

⇨ 은닉층 값들의 분포가 변화하는 양을 줄여준다

배치정규화는 z[2]_1과 x[2]_2 값이 바뀌더라도 평균과 분산은 0(or beta[2])과 1(or gamma[2]) 유지되도록 한다

- 앞선 층에서 매개변수 바뀔 때 학습하게 될 값의 분포 제한한다
- 입력값이 바뀌어서 생기는 문제를 없애고 안정화한다

2. 규제 효과

- 각각의 미니배치 X{t}가 가진 z[l]에 대해 그 미니배치의 평균과 분산에 따라 값 조정한다

- 미니배치의 데이터에서 계산한 것은 전체 데이터로부터 계산한 것 보다 잡음을 가진다. (샘플이 작아서)

=> z[l] z~[l]에서도 잡음이 있는 평균과 분산으로 계산하게 된다

- 드롭아웃에서 은닉층에 확률에 따라 0을 곱하거나 1을 곱함 (곱셈 잡음)

- 배치정규화는 표준편차로 나누고 (곱셈 잡음), 평균 뺀다 (덧셈 잡음)

=> 드롭아웃처럼 일반화 효과를 가진다

- 은닉층에 잡음을 추가하여 이후 은닉층이 하나의 은닉층에 너무 의존하지 않도록 한다

- 큰 미니배치 쓰면 일반화 효과 줄어든다

- BN은 한번에 한 미니배치에 대해 계산한다

=> 테스트 데이터에서 한번에 예시 하나씩 처리

=>테스트 과정에서는 다른 접근을 사용해야한다

<테스트시의 배치 정규화>

m : 한 미니배치 안에 샘플의 수

- $\mu = \frac{1}{m}\sum_i z^{(i)}$
- $\sigma^2 = \frac{1}{m}\sum_i (z^{(i)} - \mu)^2$
- $z^{(i)}_{norm} = \frac{z^{(i)} - \mu}{\sqrt{\sigma^2 + \epsilon}}$
- $\tilde{z}^{(i)} = \gamma z^{(i)}_{norm} + \beta$

- 테스트 과정에서는 각각 독립된 mu와 sigma^2 를 사용해야한다
- 전형적인 배치 정규화 : 여러 미니배치 거쳐서 구한 지수가중평균을 추정치로 사용
- 층 L에서 미니배치 X{1}, X{2}, .. 에 대응하k는 Y가 있다고 하면 :

  1. X{1}[l], X{2}[l], ... 각각 미니배치에 대해 **mu**를 구할 수 있다

  2. 세타1, 세타2, .. 를 구한다 (가장 최근의 평균값이 무엇인지 기록해야한다)

  => 지수가중평균이 은닉층 z값 평균의 추정치

  => 지수가중평균으로 sigma^2 추적

  mu 와 sigma^2의 이동가중평균을 구할 수 있다

  테스트 과정에서 3번째 식으로 z_norm 구할 수 있다

⇨ 테스트에서는 mu와 sigma^2를 추정할 때 지수가중 평균을 사용한다