

[Week9]_문가을

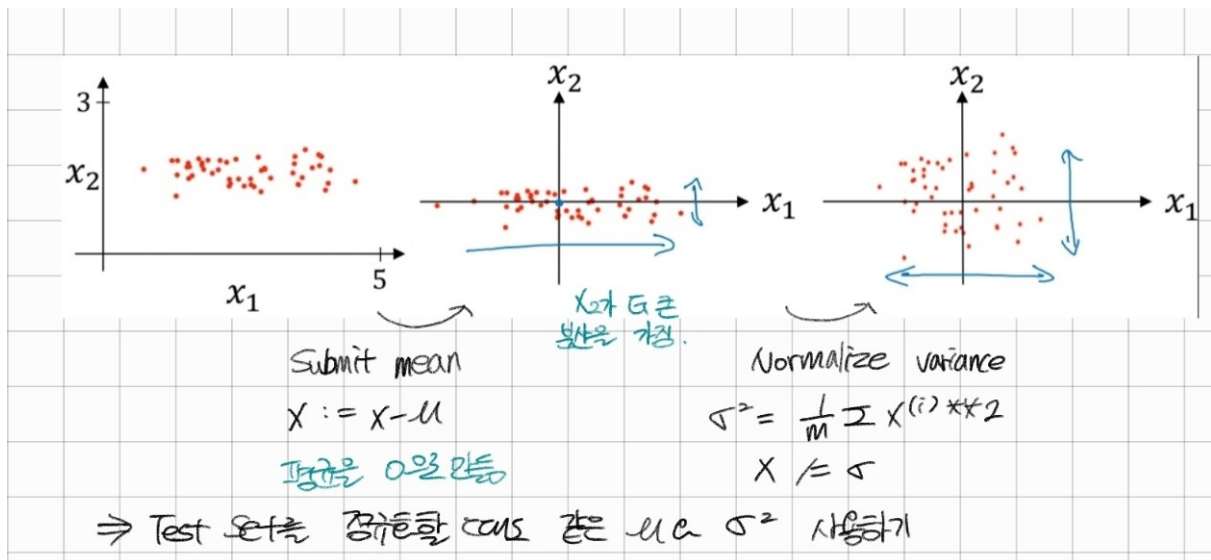
딥러닝 2단계 : 심층 신경망 성능 향상시키기

3. 최적화 문제 설정

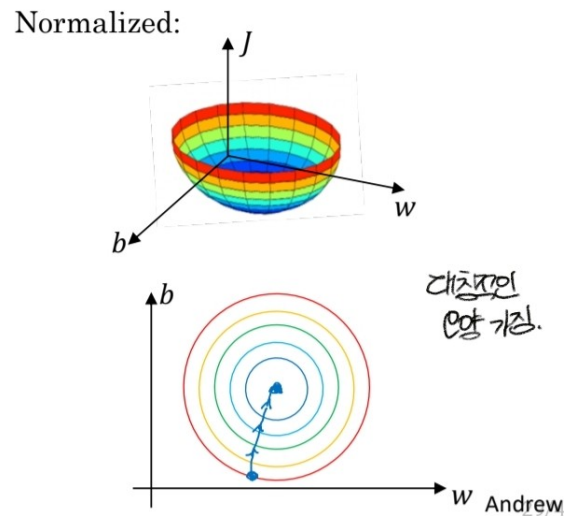
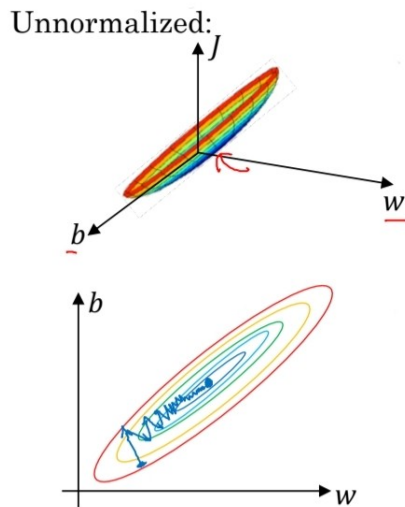
입력값의 정규화

신경망의 훈련을 빠르게 하기 위해 입력값을 정규화함.

정규화 방법



비용함수 비교



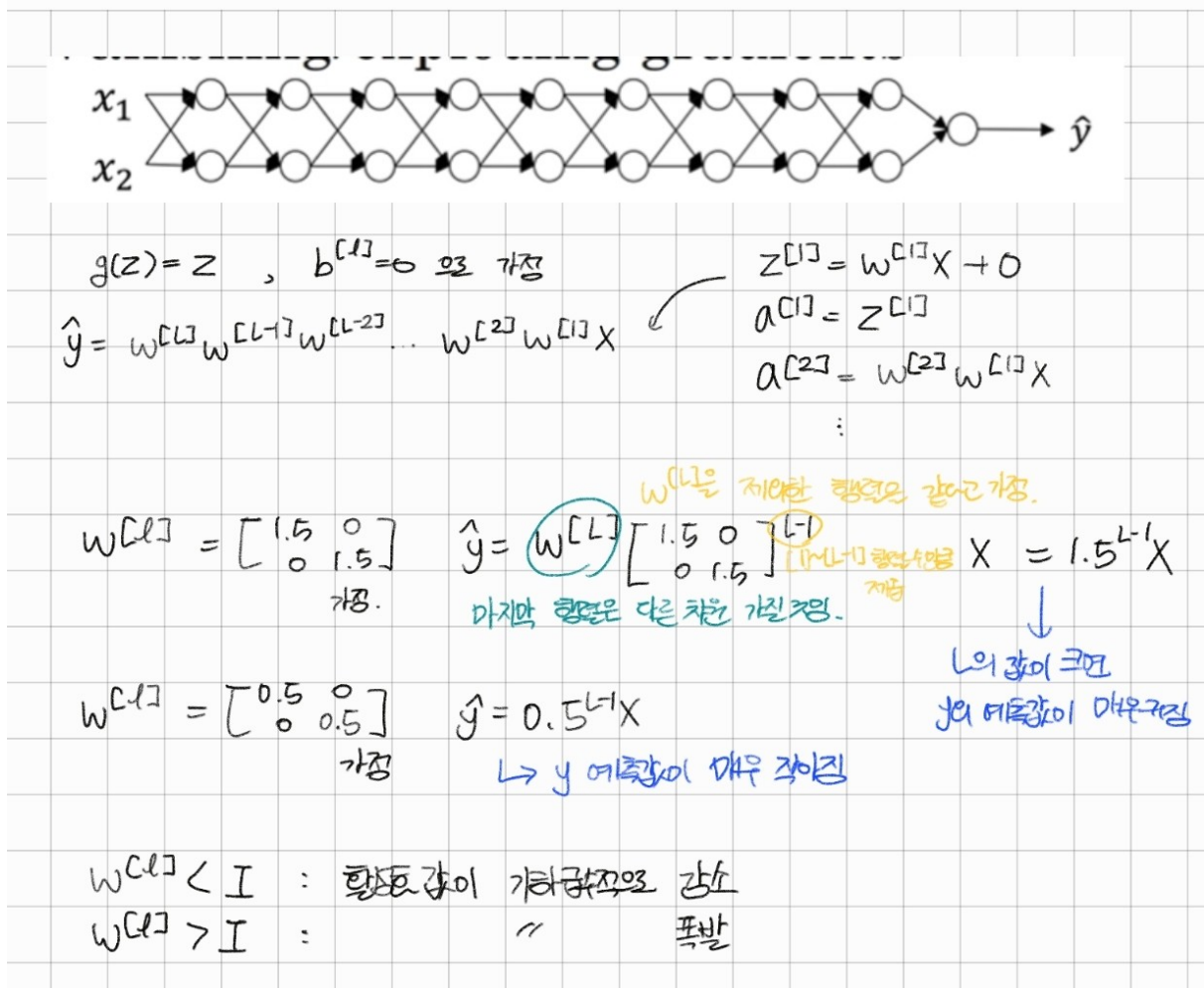
- 정규화하지 않은 데이터셋으로 훈련

→ 특성 값의 범위가 달라 매개 변수 값들이 매우 다를 것임. (그림에선 w 와 b 로 표현함.)

→ 경사하강법에서 매우 작은 학습률을 사용하게 됨. 앞뒤로 왔다갔다 하기 위해 많은 단계가 필요하기 때문.

경사 소실(vanishing gradients) / 경사 폭발(exploding gradients)

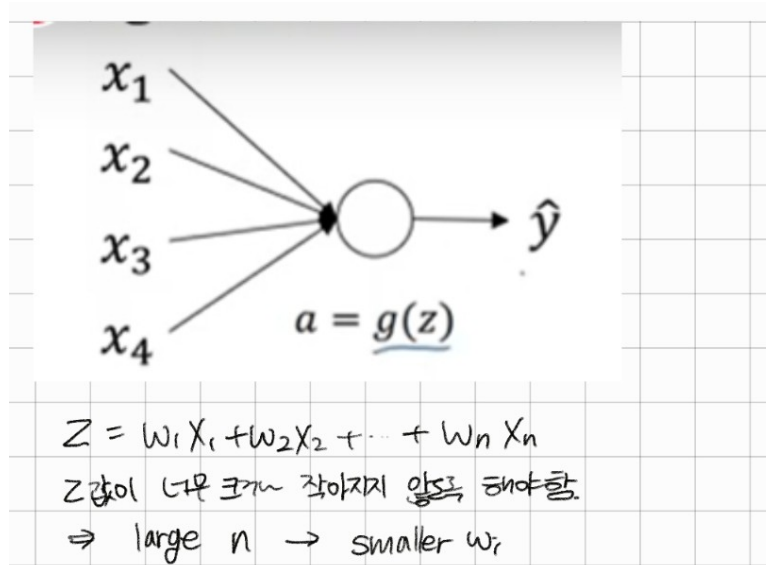
미분값이 매우 작아지거나 매우 커지는 문제.



같은 원리로 미분값이 소실/ 폭발하는 원리를 설명할 수 있음.

해결하기 위해서 가중치를 어떻게 초기화하는 지가 중요해짐.

심층 신경망의 가중치 초기화



해결 방법 : $\text{Var}(w_i) = \frac{1}{n}$ 로 설정
 $w^{[l]} = \text{np.random.randn}(\text{shape}) * \text{np.sqrt}(\frac{1}{n^{[l-1]}})$
 ReLU를 쓰면 분산은 $\frac{2}{n}$ 로 설정 층 [l]의 뉴런에 들어가는 활동의 개수의 역수

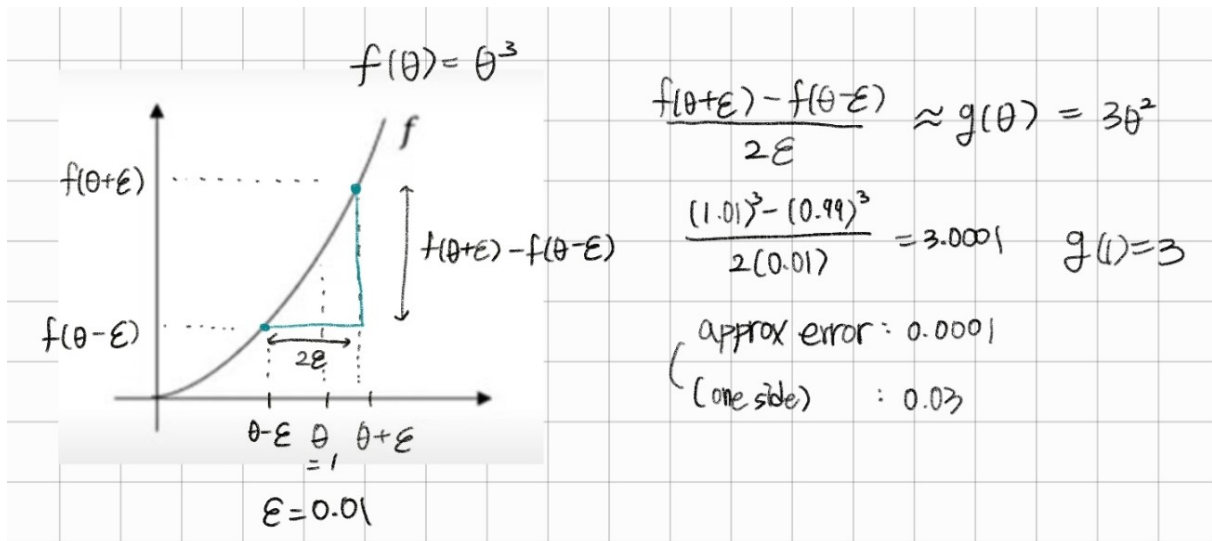
ReLU 활성화 함수를 사용하는 경우 w_i 의 분산을 $\frac{2}{n^{[l-1]}}$ 으로 설정합니다.
 tanh 활성화 함수를 사용하는 경우 w_i 의 분산을 $\frac{1}{n^{[l-1]}}$ 또는 $\frac{2}{n^{[l-1]} + n^{[l]}}$ 으로 설정합니다.

- 입력 특성 혹은 활성화값의 평균이 대략 0이고 표준편차가 1이 됨.
- z 값이 비슷한 크기를 갖게 됨.
- 경사 소실과 폭발 문제에 도움을 줌.

기울기의 수치 근사

- 역전파를 맞게 구현했는지 확인하기 위해 경사 검사를 진행함.
- 먼저 경사의 계산을 수치적으로 근사하는 방법이 필요함.

미분값 계산



왜 일반적으로 하는 미분 계산과 다른 방법을 사용하는가?

$f'(\theta) = \lim_{\epsilon \rightarrow 0} \frac{f(\theta+\epsilon) - f(\theta-\epsilon)}{2\epsilon}$ 이 근사의 오차는 $O(\epsilon^2)$
 ~~~~~ 오차가 작아짐 ~~~~~  $\epsilon = 0.01$   
 ~~~~~ 오차  $\rightarrow 0.0001$  ~~~~~ 더 작음.

$f'(\theta) = \lim_{\epsilon \rightarrow 0} \frac{f(\theta+\epsilon) - f(\theta)}{\epsilon}$ $O(\epsilon)$ $\epsilon = 0.01$
 ~~~~~ 오차  $\rightarrow 0.01$  ~~~~~

→ 오차가 작아짐.

## 경사 검사

경사 검사를 하는 방법

Take  $\underline{W}^{[1]}, \underline{b}^{[1]}, \dots, \underline{W}^{[L]}, \underline{b}^{[L]}$  and reshape into a big vector  $\underline{\theta}$ .

배열들을  $\theta$ 에 대한 배열 바꾸기. same dimension

Take  $\underline{dW}^{[1]}, \underline{db}^{[1]}, \dots, \underline{dW}^{[L]}, \underline{db}^{[L]}$  and reshape into a big vector  $\underline{d\theta}$ .

근사적인 미분값을 구하고 미분값과 비슷한지 확인하는 과정 거침.

Gradient checking (Grad check)

for each  $i$  :

$$d\theta_{\text{approx}}[i] = \frac{J(\theta_1, \theta_2, \dots, \theta_i + \epsilon, \dots) - J(\theta_1, \theta_2, \dots, \theta_i - \epsilon, \dots)}{2\epsilon}$$
$$\approx d\theta[i] = \frac{\partial J}{\partial \theta_i}$$

결과:  $d\theta_{\text{approx}}$  (vector) 나온  $\rightarrow d\theta$ 와 가까워야 함.

두 벡터가 근사적으로 같은지 확인하기 위해 유클리드 거리 구함

$$\frac{\|d\theta_{\text{approx}} - d\theta\|_2}{\|d\theta_{\text{approx}}\|_2 + \|d\theta\|_2} \approx 10^{-7} \rightarrow \text{great!}$$

$\epsilon = 10^{-7}$ 로 하면      큰 값이 나면 버그의 가능성이 있음

### 경사 검사 시 주의할 점

#### 1. 디버깅을 위해서만 경사 검사를 사용하기

$\rightarrow$  근사 미분값을 구하는 데에 시간이 오래 걸리기 때문임.

#### 2. 경사 검사의 알고리즘이 실패하면 ( 근사 미분값과 미분값의 차이가 크면 )

$\rightarrow$  개별적인 컴포넌트를 확인해 버그를 확인하기.

$\rightarrow$  각각  $i$ 에 대해  $d\theta_{\text{approx}}[i]$ 와  $d\theta[i]$  확인

#### 3. 비용함수에 정규화 term이 있다는 것을 기억하기

- $J(w, b) = \frac{1}{m} \sum_{i=1}^m \ell(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m} \|w\|_2^2$
- $L_2$  regularization:  $\|w\|_2^2 = \sum_{j=1}^{n_x} w_j^2 = w^T w$
- $L_1$  regularization:  $\frac{\lambda}{2m} \|w\|_1 = \frac{\lambda}{2m} \sum_{j=1}^{n_x} |w_j|$

4. 드롭아웃에서는 경사 검사가 작동하지 않는다.

→ 드롭아웃에서 비용함수를 계산하기 어렵기 때문임.

5. 무작위적 초기화에서  $w$ 와  $b$ 가 0에 가까울 때 경사 검사가 잘 되는 경우

→ 훈련을 조금 시켜  $w$ 와  $b$ 가 0에서 멀어지게 한 다음 경사 검사를 다시 해보기