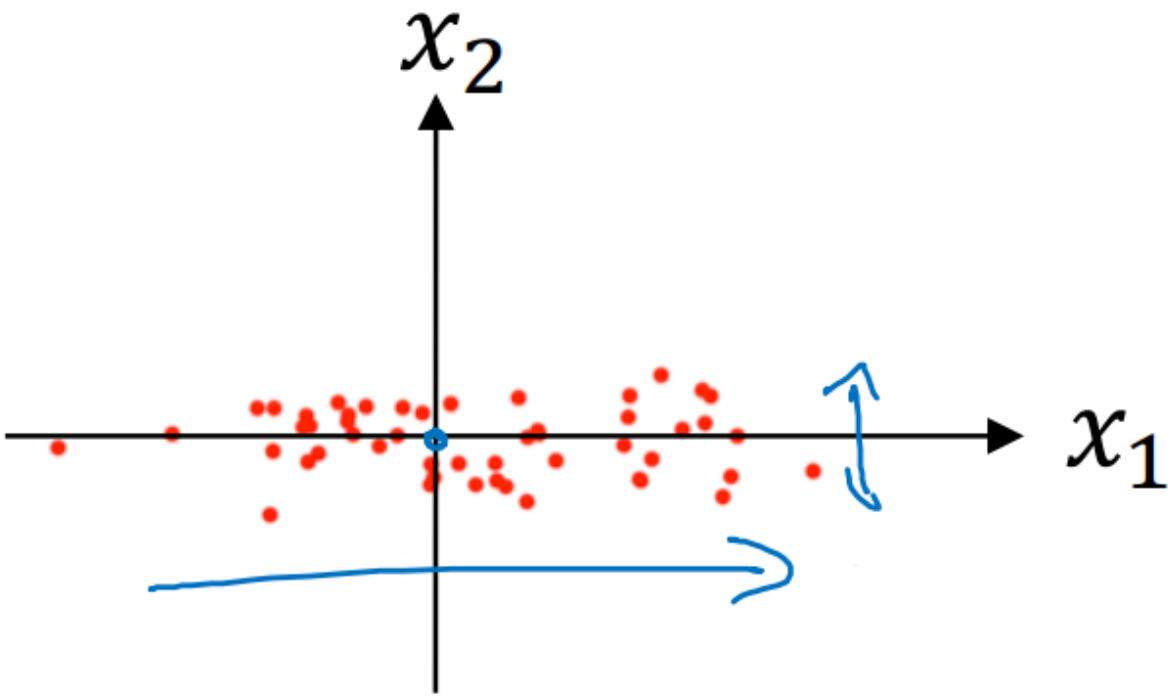
[딥러닝 2단계] 3. 최적화 문제 설정

Normalization

- 훈련을 빠르게 할 수 있는 기법이다.
- 입력을 정규화한다.

두 가지 방법이 있다.

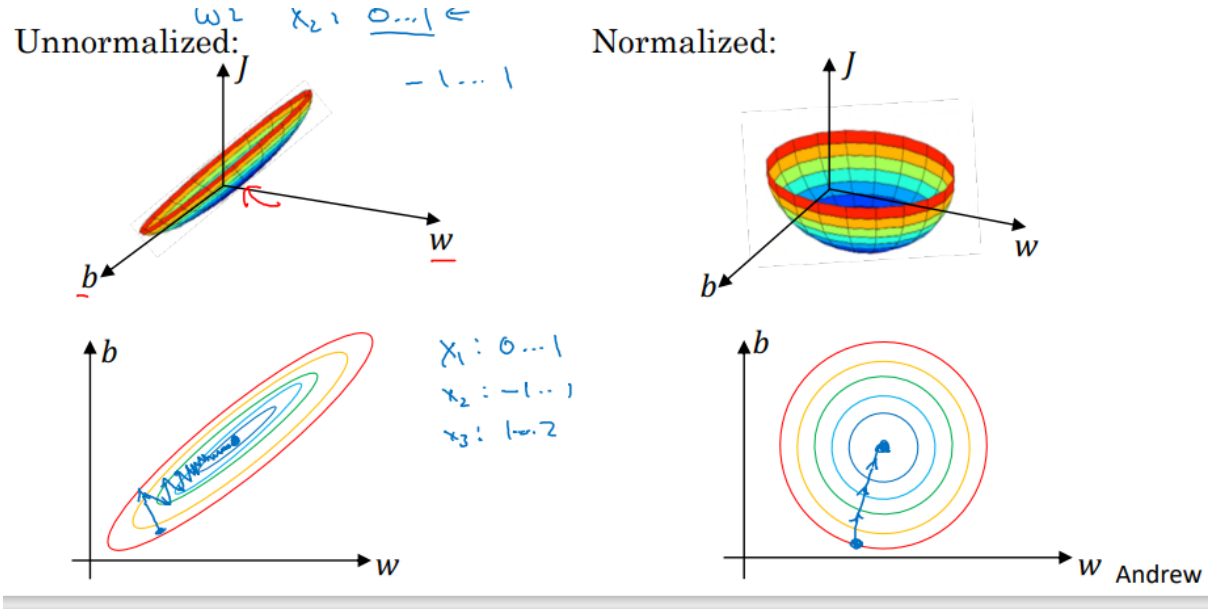
1. 입력 데이터의 평균이 0이 되도록 모든 값마다 평균을 뺀다.



2. 분산을 1로 만들어 각 특성마다 같은 분산을 가지게 한다.

훈련 데이터를 확대할 때 사용한다면, 테스트 세트에도 같은 μ와 σ를사용하여 똑같이 정규화해야 한다.

Normalization이 필요한 이유



(좌) Normalize O, (우) Normalize X

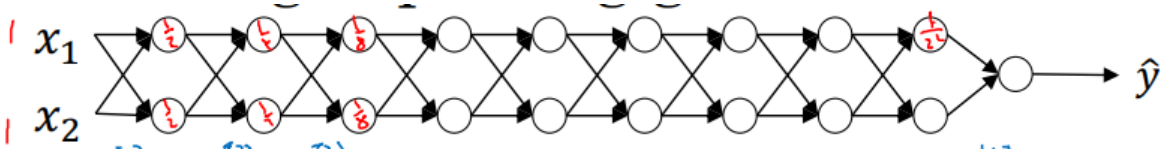
입력 특성들의 분포가 다른데 정규화되지 않았다면, 가늘고 긴 모양의 비용함수를 얻는다. **경사 하강법 실행 시 매우 작은 학습률이 필요**하다.

입력 특성의 분포를 정규화하면 원 모양의 비용함수를 얻는다. 경사 하강법 실행 시 **큰 학습률을 가져도** 최솟값을 찾을 수 있다.

- 정규화는 어떤 해도 가하지 않기 때문에 되도록 하는 것을 추천한다.

경사소실 & 경사폭발

매우 깊은 신경망을 훈련 시, 미분값이 매우 작아지거나 매우 커지는 문제 발생



예시) 두 은닉 유닛만을 가진 매우 깊은 신경망

선형 활성화 함수를 사용한다고 가정하면, y = w^[l] * … * w^[2] * w^[1] * x의 형태가 된다. 즉, 모든 행렬들을 곱한 값이다.

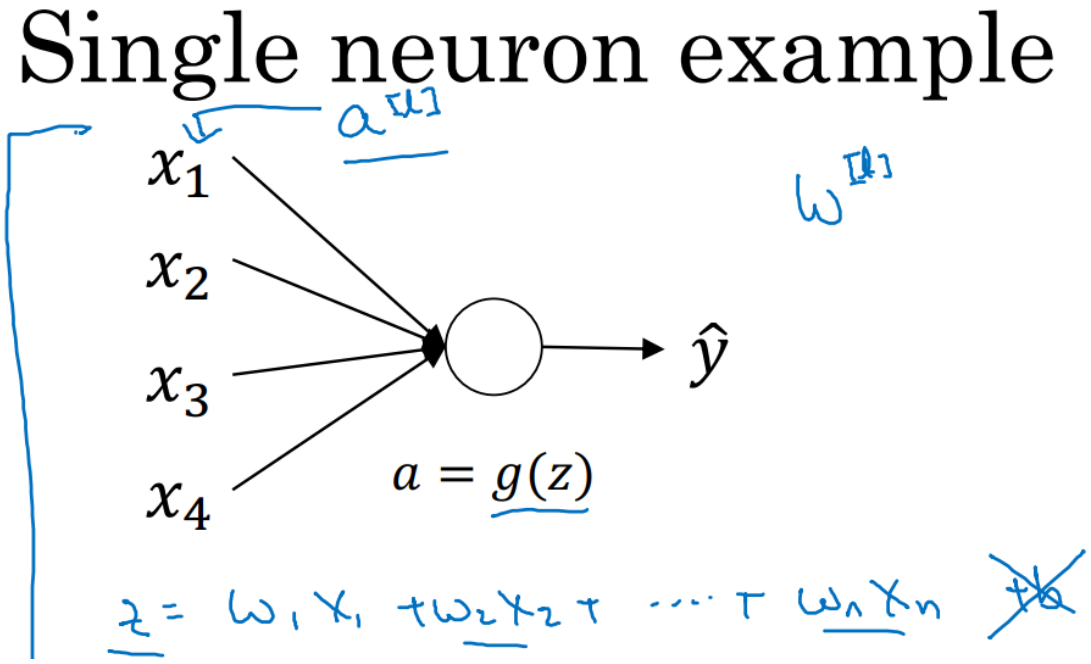
만약 w > 1이면, y의 예측값은 x * w^(l-1)이 되어 매우 커지고, 이를 폭발이라고 한다.

만약 w < 1이면, y의 예측값은 매우 작아지고, 이를 소실이라고 한다.

특히 경사가 매우 작은 경우에 학습시키는 데 매우 오랜 시간이 걸려 학습이 어려워진다.

심층 신경망의 가중치 초기화

신경망 가중치 초기화를 섬세하게 하는 방법을 통해 경사 소실과 폭발의 문제를 부분적으로 해결할 수 있다.



z의 값이 너무 크거나 작아지지 않도록 해야 한다.

- 가중치 초기화 방법

1. w_i 의 분산을 $\frac{1}{n}$ 으로 설정합니다. (n : 입력 특성의 개수)
2. ReLU 활성화 함수를 사용하는 경우 w_i 의 분산을 $\frac{2}{n^{[l-1]}}$ 으로 설정합니다.
3. tanh 활성화 함수를 사용하는 경우 w_i 의 분산을 $\frac{1}{n^{[l-1]}}$ 또는 $\frac{2}{n^{[l-1]}+n^{[l]}}$ 으로 설정합니다.

1. w[l] = np.random.randn(shape) * np.sqrt(1/n^[l-1])

- n^[l-1] : 층 l 뉴런의 특성 개수

2. ReLU를 이용하는 경우, 분산을 2/n으로 설정하는 것이 더 잘 작동한다.

→ 이 방법은 가중치 w의 범위를 제한한다.

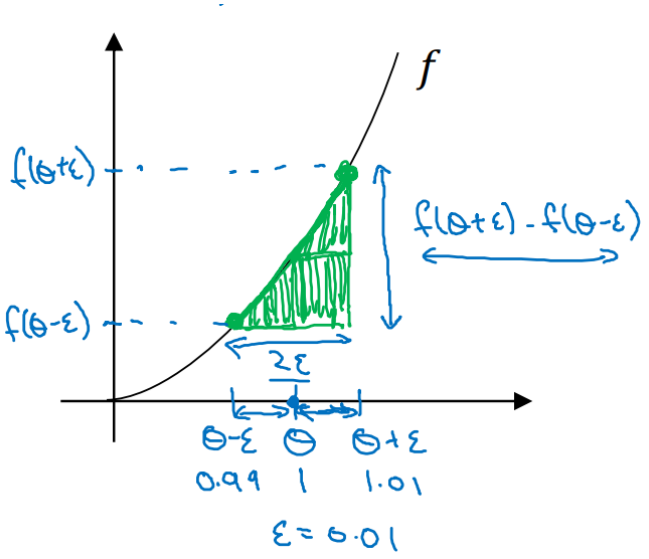
3. tanh를 이용하는 경우, 상수 2 대신 1 사용 = 세이비어 초기화

분산 매개변수도 하이퍼파라미터이다. 그러나 다른 하이퍼파라미터보다는 중요성이 떨어진다.

기울기의 수치 근사

경사 검사 테스트

- **역전파를 알맞게 구현했는지 확인하는 방법**이다.
- 경사를 수치적으로 어떠한 값에 근사한 후, 경사 검사를 실시해 역전파 구현이 맞는지 확인한다.



1. θ에서 좌우로 ε만큼 이동해 f(θ-ε)와 f(θ+ε)를 얻는다.
2. 밑변이 2ε인 삼각형에서 높이/너비 값을 계산한다.
    a. 더 큰 삼각형에서 높이/너비 값을 계산하는 것이 도함수 근사에 더 좋은 값을 준다.
    b. 높이/너비 값 = (θ+ε)-f(θ-ε) / 2ε
3. 이 값을 계산해보면 g(θ)와 비슷한 것을 확인할 수 있다.
    a. 한 쪽의 차이만을 이용하는 것보다 두 배는 느리게 실행되지만, 훨씬 정확한 값을 얻을 수 있다.

도함수의 정의

1. 양쪽의 차이 사용

$$f'(\theta) = \lim_{\varepsilon \to 0} \frac{f(\theta+\varepsilon) - f(\theta-\varepsilon)}{2\varepsilon}$$

- 0이 아닌 ε에 대해서 이 근사의 오차는 O(ε^2)이다.

2. 한 쪽의 차이 사용

- 0이 아닌 ε에 대해서 이 근사의 오차는 O(ε)이다.

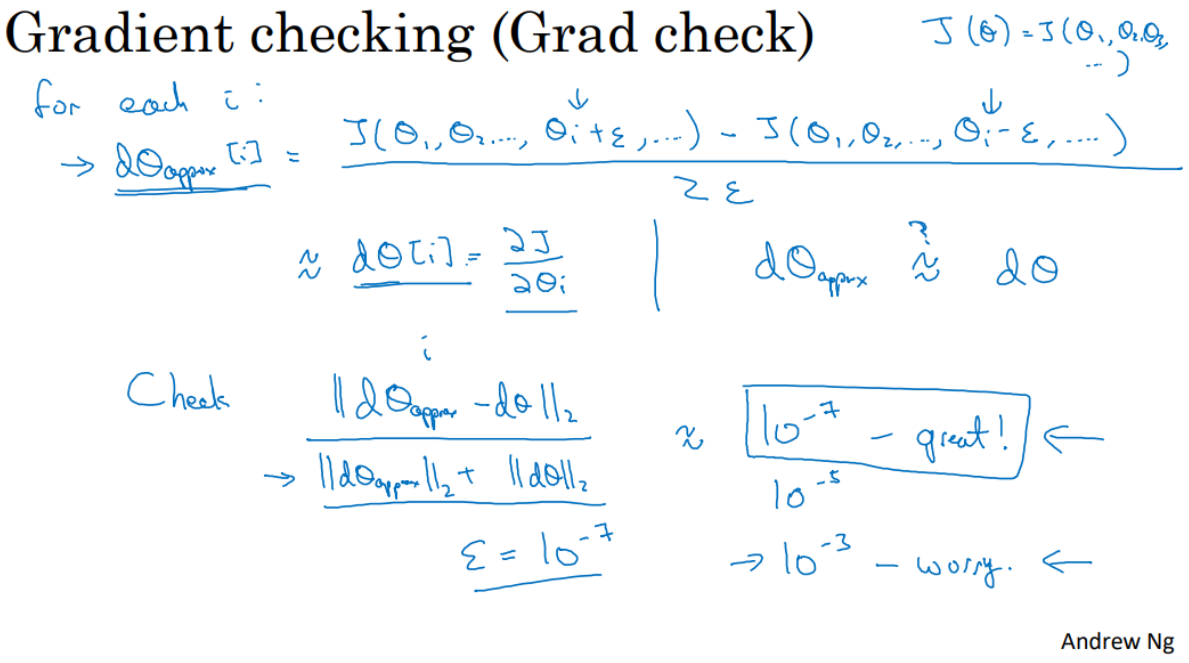
ε은 0과 1 사이의 매우 작은 수이므로, O(ε^2)는 O(ε)보다 훨씬 작다.

따라서 경사 검사 시 양쪽의 차이를 사용하는 첫 번째 방법을 사용하는 것이 훨씬 좋다.

경사 검사(Gradient Checking)

역전파를 알맞게 구현했는지 확인하는 방법

1. 모델 안에 있는 모든 변수(W, b)를 하나의 벡터(θ)로 concatenate 한다.
2. 모델 안에 있는 모든 derivative(dW, db)를 하나의 벡터(dθ)로 concatenate 한다.
3. 비용 함수는 J(W, b) 에서 J(θ) 로 변한다.



Andrew Ng

4. 수치 미분을 구한다.

$$d\theta^{[i]}_{approx} = \frac{J(\theta_1, \cdots, \theta_i + \epsilon, \cdots) - J(\theta_1, \cdots, \theta_i - \epsilon, \cdots)}{2\epsilon}$$

- 이 값은 d(*θ*[i])와 근사적으로 같아야 한다. (비용함수의 도함수)
- 즉, 아래 식이 맞는지 확인해야 한다.

$$d\theta^{[i]}_{approx} \approx d\theta$$

5. 유클리디안 거리를 사용하여 두 벡터 간의 유사도를 계산한다. L2 norm을 이용한다.

$$\frac{\|d\theta^{[i]}_{approx} - d\theta\|_2}{\|d\theta^{[i]}_{approx}\|_2 + \|d\theta\|_2}$$

- 교수님은 **ε을 10^-7로 사용**하신다. 이때, 유클리디안 거리에 따라 결과를 파악할 수 있다.
  - 10^-7 이하: 근사가 매우 잘 되었다!
  - 10^-5: 괜찮은 값이지만, 벡터의 원소를 이중으로 확인하여 너무 큰 원소가 있는지 확인한다. 원소 간의 차이가 너무 크면 코드에 버그가 있을 수 있다.
  - 10^-3: 버그 가능성이 매우 높다. θ의 개별 원소를 신중히 살피며 특정 i에 대해 dθapprox[i]와 dθ[i]의 차이가 심한 값을 추적한다. 미분 계산이 틀린 곳이 있는지 확인해야 한다.

경사 검사 시 주의할 점

- 모든 i의 값에 대한 dθapprox[i]를 계산하는 부분의 속도가 매우 느리기 때문에, **훈련에서는 경사 검사를 절대 사용하면 안 된다. 디버깅 시에만 활용한다.**
- 알고리즘이 경사 검사에 실패 했다면, 어느 원소 부분에서 실패했는지 찾아봐야 한다. 특정 부분에서 계속 실패한다면, 그 경사가 계산된 층에서 문제가 생긴 것을 확인할 수 있다.
  - θ값이 b에는 맞지 않고, w와 유사할 경우, db를 어떻게 계산하느냐에 따라 버그가 발생할 것이다.

$d\theta$ 는 θ 에 대응하는 J의 정규화 항($\frac{d}{dm}\sum_m \|w^{[l]}\|_F^2$)도 포함하기 때문에 경사 검사 계산시 같이 포함해야합니다.

- 경사 검사는 드롭아웃에서는 작동하지 않는다. 매번 무작위로 노드를 삭제하기 때문에, 드롭아웃을 이용한 계산을 이중으로 확인하기 위해 경사 검사를 사용하기는 어렵다.

→ **훈련 후 경사 검사를 먼저 한 후, 드롭아웃을 진행한다.**

- 마지막으로 거의 일어나지 않지만 가끔 무작위 초기화를 해도 초기에 경사 검사가 잘 되는 경우, w와 b가 0에 가까워서일 수 있다. 이때는 무작위 초기화 상태에서 훈련을 잠시 시킨 후 경사 검사를 재실행한다.