9주차

태그	완료

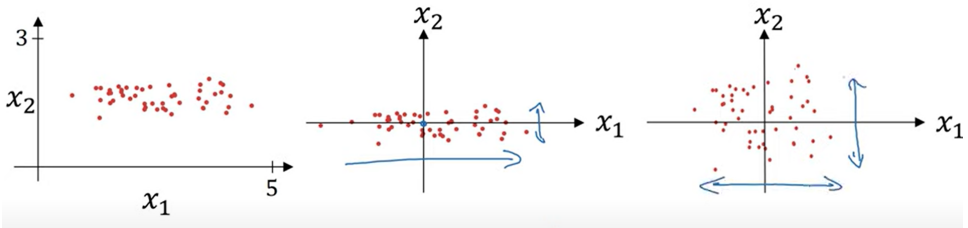
Normalizing Inputs

- 입력의 정규화 : 신경망의 훈련을 빠르게 할 수 있는 하나의 기법
- 방법
  1. substract mean : 평균을 0으로 만든다.

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}, x = x - \mu$$

  2. normalize variance : 분산을 1로 만든다.
     - σ^2는 각 특징의 분산에 대한 벡터, 이미 $x^{(i)} - \mu$를 했기 때문에 편차의 제곱으로 계산된다.
     - x1와 x2의 분산은 모두 1과 같아진다.

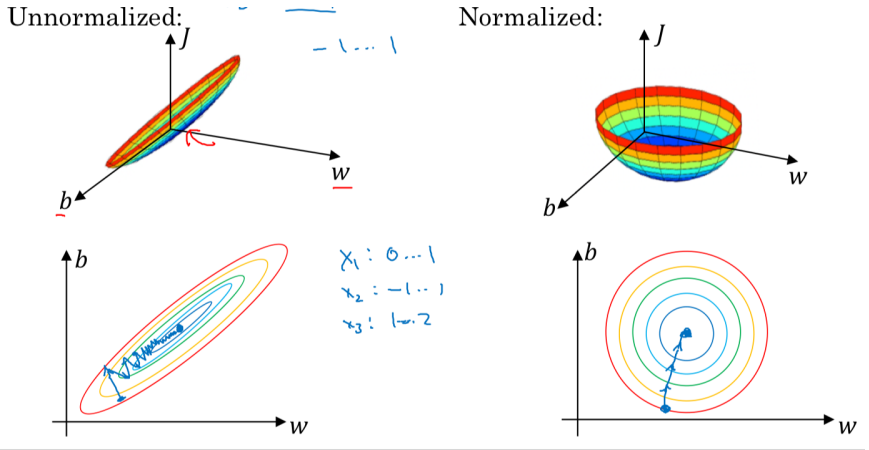
$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} * *2, x = \frac{x}{\sigma}$$



original data 1. substract mean 2. normalize variance

- 훈련 세트와 테스트 세트를 정규화할 때 같은 μ와 σ를 사용한다.

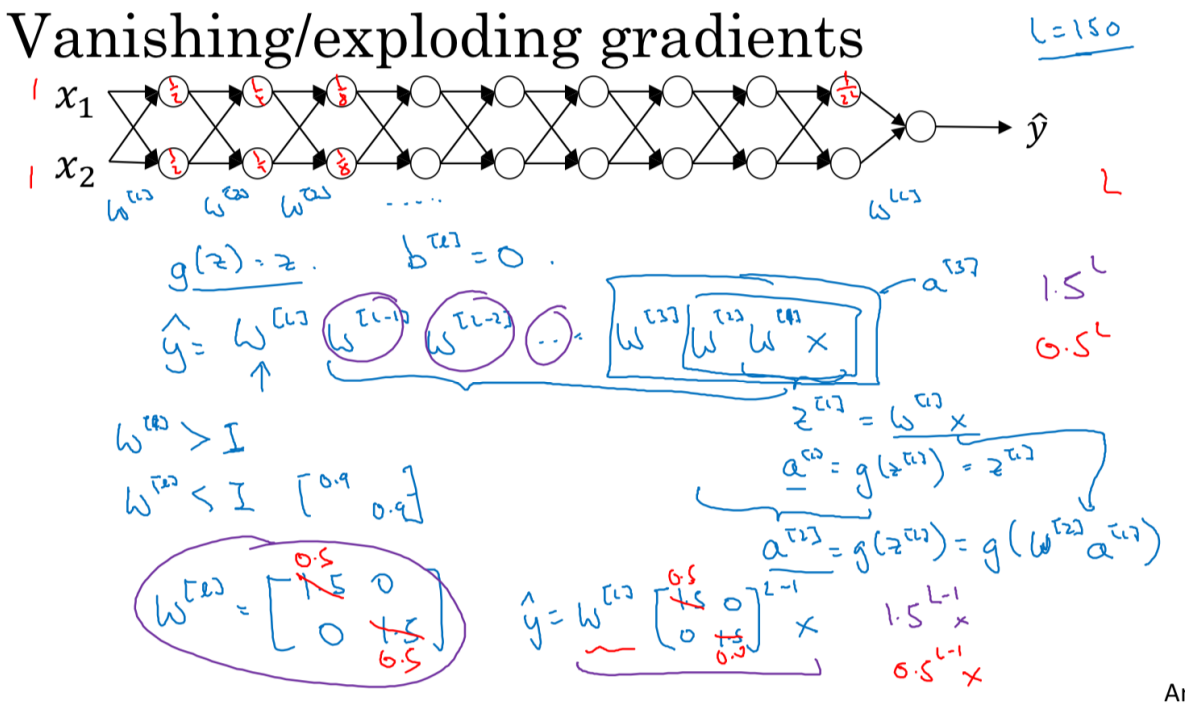
- Why normalize inputs?
  - 만약 특성들이 매우 다른 크기(scale)를 갖고 있다면 매개변수에 대한 값이 범위가 굉장히 다른 값을 갖게 될 것이기 때문이다.
  - 예를 들어 x1 : 1~1000이고 x2 : 0~1이라면 비용 함수가 왼쪽과 같은 그래프와 같이 길쭉한 모양으로 그려질 것이다. 또한 gradient descent를 할 때 learning rate가 작아야 한다.
  - 그러나 오른쪽과 같이 특징들이 비슷한 크기를 갖도록 정규화를 거치면 비용 함수가 둥글고, 대칭적, 최적화하기 쉽게 그려진다. 이로 인해 학습 알고리즘이 빨리 실행된다.



- 만약 특징들의 크기가 비슷하다면 정규화를 하지 않아도 괜찮지만, 일반적으로 정규화는 학습에 아무런 해도 끼치지 않기 때문에 하는 것을 추천한다.

경사소실/경사폭발

- 매우 깊은 신경망을 학습시킬 때 발생하는 문제이다.

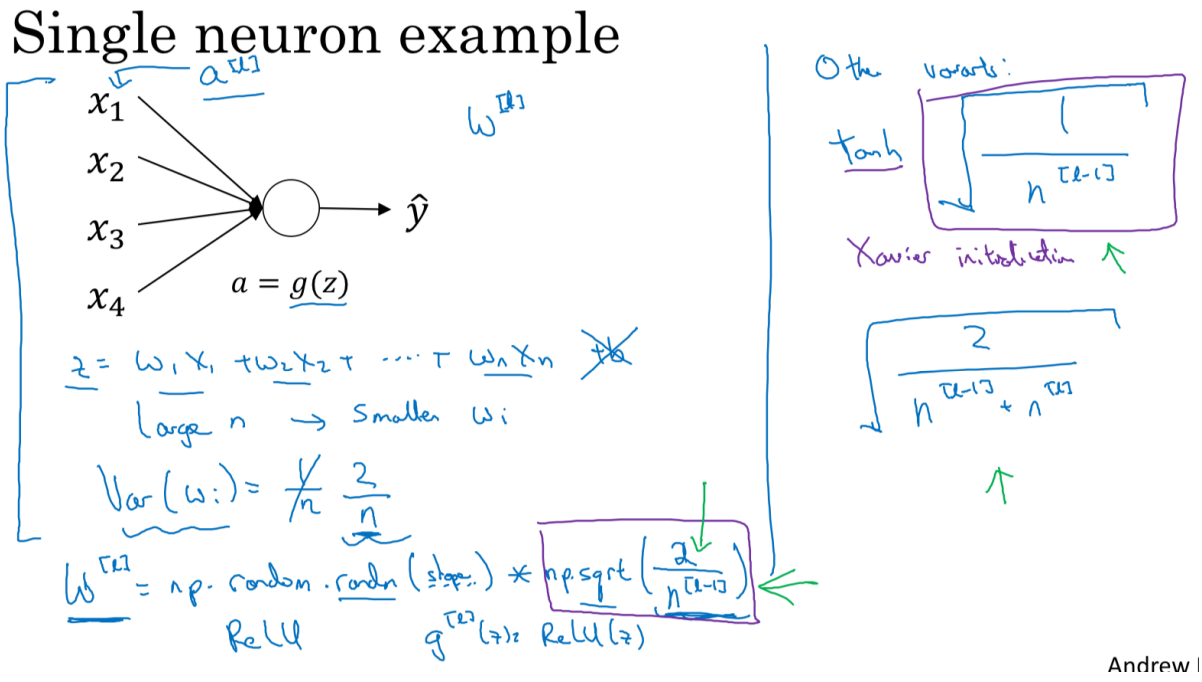


AI

- 네트워크의 깊이를 L이라고 가정한다. 활성화 함수 g가 g(z) = z (선형 함수)라고 가정한다. $b^{[l]} = 0$이라고 가정한다.
- $\hat{y} = a^{[L]} = w^{[l]} w^{[l-1]} \dots w^{[2]} w^{[1]} x$이 된다.
- 이때 가중치 행렬 $w^{[l]}$가 1.5로 단위 행렬보다 크면, L이 깊어질 때 $\hat{y} = a^{[L]} = 1.5^L x$로 기하급수적으로 커질 것이다. → 경사 폭발
- 0.5로 단위 행렬보다 작다면 L이 깊어질 때 $\hat{y} = a^{[L]} = 0.5^L x$로 기하급수적으로 작아질 것이다. → 경사 소실
- 경사 폭발과 경사 소실로 인해 학습하는 데 많은 시간이 걸린다. → 가중치 초기값을 신중하게 결정해야 한다.

심층 신경망의 가중치 초기화

- 가중치 행렬이 1보다 너무 크거나 작지 않도록 설정해서, 경사폭발과 경사소실이 일어나지 않도록 해야 한다.
- 가중치 초기화 방법

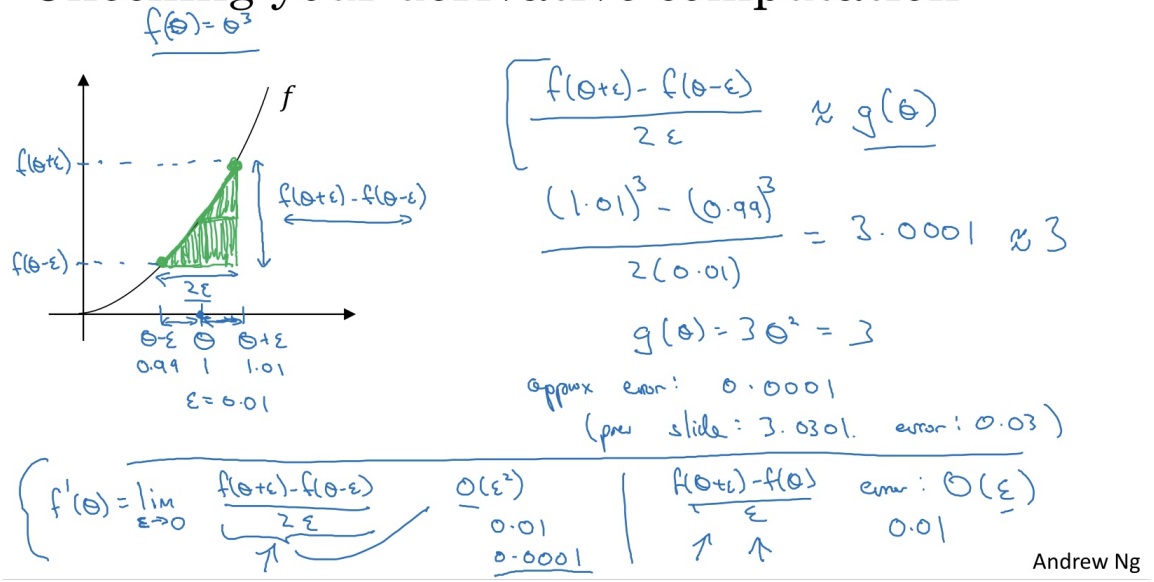


- n이 입력 특성의 개수일 때, z가 너무 크지 않도록 해야 하므로 n이 커질수록 w_i의 값이 작아져야 한다.
- 초기 가중치 행렬 w_i에 대한 분산 : $Var(w_i) = \frac{1}{n}$
  - $w^{[l]}$ = np.random.randn(shape)*np.sqrt($\frac{2}{n^{[n-1]}}$) → l번째 층의 각각의 노드에 들어가는 입력 수는 $n^{[l-1]}$개이다.
  - 활성화 함수 ReLU : 분산으로 $\frac{1}{n^{[l-1]}}$보다$\frac{2}{n^{[l-1]}}$을 사용한다.
  - 활성화 함수가 tanh라면 $\frac{2}{n^{[l-1]}}$보다$\frac{1}{n^{[l-1]}}$ 또는 $\frac{2}{n^{[l-1]}+n^{[l]}}$을 사용다. 이것을 세이비어 초기화라고 한다.

기울기의 수치 근사

- 경사 검사 : 역전파를 올바르게 구했는지 확인

  → 경사의 계산을 수치적으로 구해야 한다.

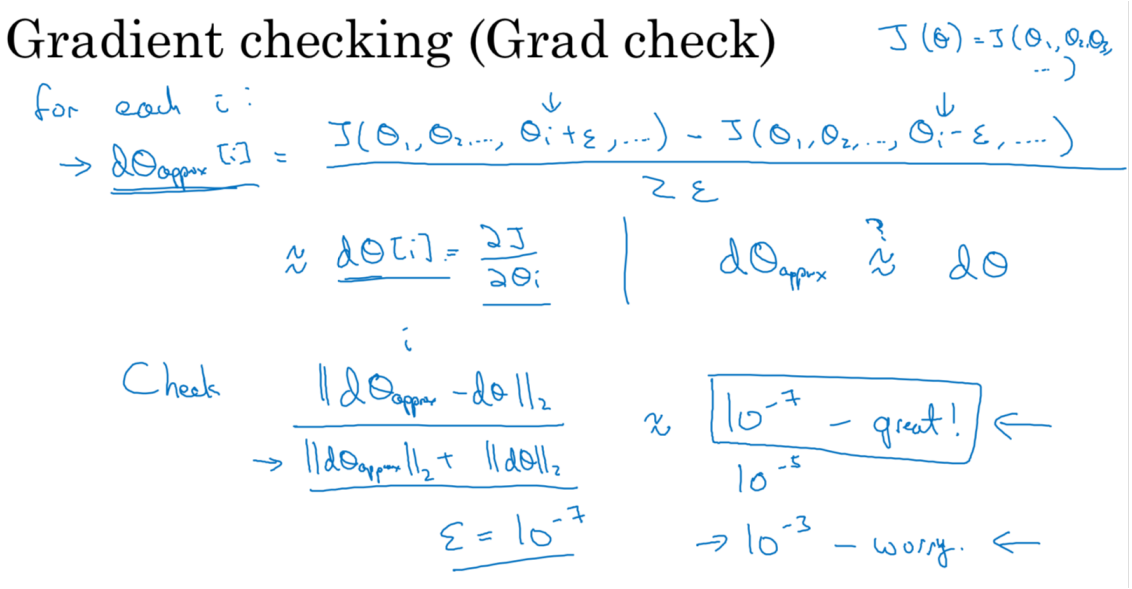


- $f(\theta) = \theta^3$이라고 하고, 그 도함수를 $g(\theta)$라고 한다. ε은 매우 작은 수이다.
- θ와 $\theta + \varepsilon$의 기울기를 구하는 전향 차분 근사보다, $\theta + \varepsilon$과 $\theta - \varepsilon$의 기울기를 구하는 중앙 차분 근사가 θ에서의 $f(\theta)$의 기울기, $g(\theta)$에 더 잘 근사한다.
- 실제로 전향 차분 근사를 사용하면 error는 0.03이지만 중앙 차분 근사를 사용하면 error는 0.0001에 그친다.
- 전향 차분 근사에서의 절단 오차는 O(ε)이고 중앙 차분 근사에서의 절단 오차는 O(ε^2)이다. 따라서 중앙 차분 근사에서의 오차가 훨씬 작은 것이다.

경사 검사

- 경사 검사 : 시간을 절약 & 역전파 구현에 대한 버그를 찾음
- 방법
  1. $W^{[1]}, b^{[1]} \dots W^{[L]}, b^{[L]}$를 매우 큰 벡터 θ로 reshape한 후 모두 concatentate 한다. → 비용함수를 (W, b)가 아닌 θ에 대한 함수 $J(\theta)$로 나타낸다.
  2. $dW^{[1]}, db^{[1]} \dots dW^{[L]}, db^{[b]}$를 매우 큰 벡터 $d\theta$로 reshape한 후 모두 concatenate 한다.

  → 이때, 각각의 대응되는 가중치 행렬/편향과 그 미분값은 같은 차원을 갖는다.
- Gradient Checking (Grad check)



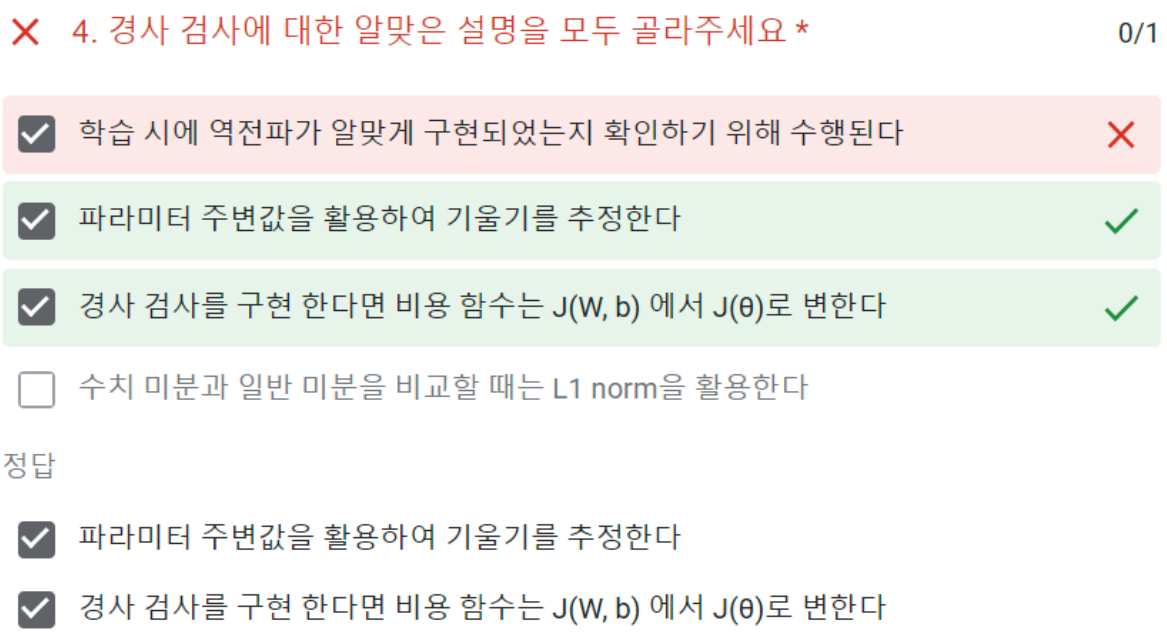
- $J(\theta) = J(\theta_1, \theta_2, ...\theta_i...\theta_n)$이다. i에 대해 반복문을 돌린다.
- $d\theta_{approxi}[i]$를 중앙 차분 근사를 사용해 구한다. 이때 θ_i에만 ε을 더하고 빼준다.
- 수치 미분과 일반 미분의 비교 : $d\theta_{approxi}[i]$는 $d\theta[i]$(=$\frac{dJ}{d\theta_i}$) 근사한다.
- $d\theta_{approxi}$와 $d\theta$의 차원은 모두 θ로 같다. 이때 이 두 값의 유사도를 확인해야 한다.
  - Euclidean distance를 통해 계산할 수 있다.
  - 이때 분자는 뺀 값의 L2 norm을 이용해 구하고, 분모는 각각의 L2 norm을 더한 것으로 설정해 정규화까지 해준다.
  - 만약 이 값이 10^{-7}보다 작다면 매우 근사가 잘 되었음을 의미한다. 10^{-5}라면 더 자세히 살펴볼 필요가 있다. 10^{-3}이라면 우려할 만한 사항으로, 버그의 가능성이 크다. → 값이 큰 i에 대해 미분이 잘못 되었는지 확인해야 한다.

경사 검사 시 주의할 점

- 경사 검사는 속도가 매우 느리기 때문에 훈련에 사용 x, 디버깅할 때만 사용한다.
- 만약 알고리즘이 경사 검사에 실패한다면, 개별적인 원소를 확인해 버그를 확인하라.
  - 만약 $d\theta_{approxi}$가 $d\theta$와 매우 유사하지 않을 경우, 각각의 i에 대하여 이런 문제가 발생하는지 확인한다.
- Regularization을 하라.
  - 비용 함수 $J(\theta)$의 도함수 $d\theta$는 frobenius norm의 제곱의 합, 즉 정규화 항을 포함한다.
- 경사 검사는 dropout에서 작동하지 않는다.

- 매번 다른 부분집합의 노드를 무작위로 삭제하기 때문에 비용 함수를 구현하기 어렵다.
- 통상적으로 드롭아웃을 끄고 알고리즘이 최소한 드롭아웃 없이 맞는지 확인하고, 경사 검사를 사용한 뒤 다시 드롭아웃을 킨다.
- (드물게 발생) 무작위 초기화에서 w와 b가 0에 가까울 때, 즉 경사 하강법의 구현이 올바르게 된 경우와 같을 때 → 그러나 경사 하강법이 구현되면 w와 b가 커진다. 좋지 않음 ! → 무작위 초기화에서 경사 검사를 실행, 네트워크를 조금 훈련해 w와 b가 0에서 멀어지게 한 후 다시 경사 검사를 실행한다.

Quiz



- 경사 검사는 시간이 너무 오래 걸리기 때문에 '학습 시'에는 사용되지 않고 디버깅 때만 사용된다.