

2. 신경망 네트워크의 정규화

정규화

정규화가 오버피팅을 막는 이유

Dropout Regularization

Understanding Dropout

다른 정규화 방법들

정규화

- 정규화는 오버피팅을 막고 분산을 줄여줌

L2 norm

$$J(w, b) = \frac{1}{n} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m} \|w\|_2^2 \quad (\lambda: \text{regularization param})$$

$$\bullet \text{ L2 정규화: } \|w\|_2^2 = \sum_{j=1}^{n_y} w_j^2 = w^T w$$

$$\bullet \text{ L1 norm: } \frac{\lambda}{m} \sum_{j=1}^{n_y} |w_j| = \frac{\lambda}{2m} \|w\|_1 \rightarrow w \text{의 0이 많아짐} \\ \Rightarrow \text{사용 많이 } (\lambda)$$

Frobenius norm

$$J(w^{[1]}, b^{[1]}, \dots, w^{[L]}, b^{[L]}) = \frac{1}{n} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m} \sum_{l=1}^L \|w^{[l]}\|^2$$

$$\bullet \text{ Frobenius norm} \\ \|w^{[l]}\|_F^2 = \sum_{i=1}^{n_{l-1}} \sum_{j=1}^{n_l} (w_{ij}^{[l]})^2$$

weight decay

- L2 norm을 사용하면 back propagation에서 항상 weight이 감소함

• 경사하강법

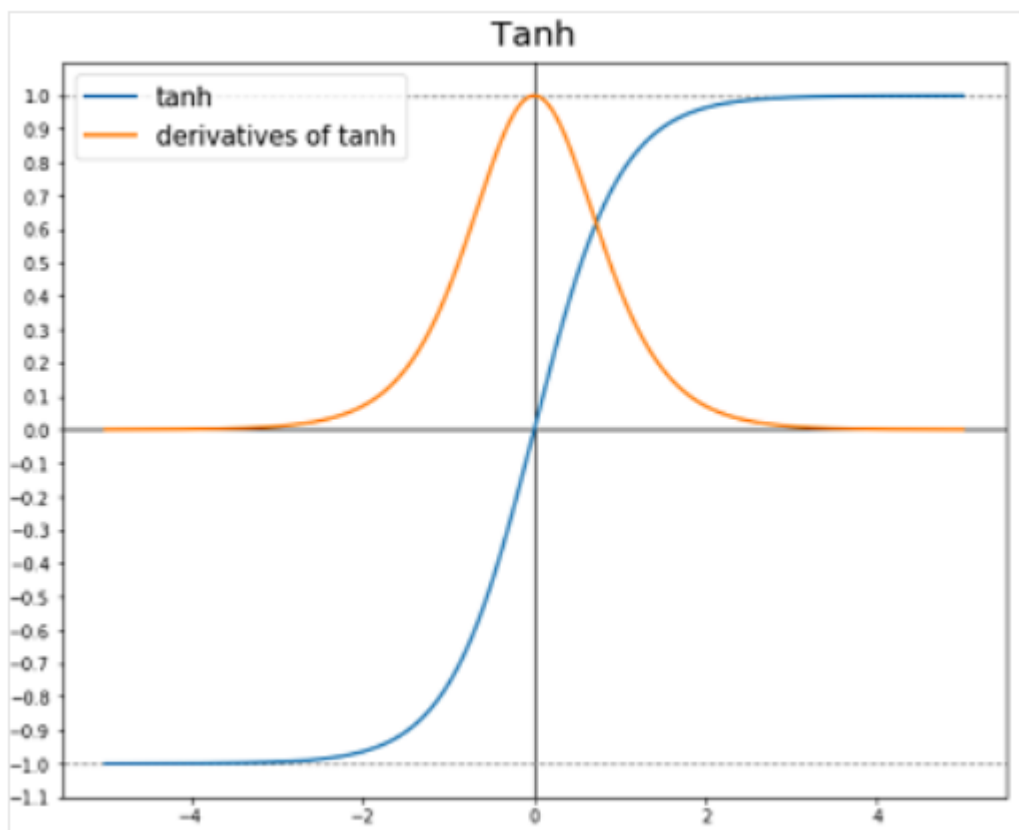
$$dW^{[L]} = (\text{from backprop}) + \underbrace{\frac{\lambda}{m} W^{[L]}}$$

$$\rightarrow W^{[L]} := W^{[L]} - \alpha dW^{[L]} \quad \text{"weight decay" (가중치 감소)}$$

$\Rightarrow W^{[L]}$ 이 어떤 값이든 값이 작아짐

정규화가 오버피팅을 막는 이유

- L2 norm에서 lambda의 값을 크게 만들면 가중치 w가 0에 가깝게 됨 \rightarrow 간단하고 작은 신경망이 되어 오버피팅이 덜 일어난다.
- tanh 함수에서 lambda가 커지면 w가 작아지고, w가 작아지면 z가 작아짐
- tanh 함수에서 z가 작을 때 $g(z)$ 가 선형에 가깝게 되고, 전체 네트워크도 선형이 되기 때문에 모델의 복잡도가 줄어 오버피팅 방지됨



Dropout Regularization

- 신경망의 각 층에서 노드를 삭제할 확률을 설정
- 확률에 따라 노드를 삭제해 네트워크를 감소시켜 학습 진행

dropout 구현 방법

- inverted dropout
 - keep_prob : 노드가 삭제되지 않을 확률
 - keep_prob값을 다시 나눠줌으로써 기댓값을 1로 유지
 - test 할 때는 no dropout
 - test할 때 스케일링 변수를 추가하지 않아도 됨

layer $l=3$, keep-prob = 0.8

$d3 = \text{np.random.rand}(a3.\text{shape}[0], a3.\text{shape}[1]) < \text{keep-prob}$

$a3 = \text{np.multiply}(a3, d3)$

$a3 /= \text{keep-prob}$

Understanding Dropout

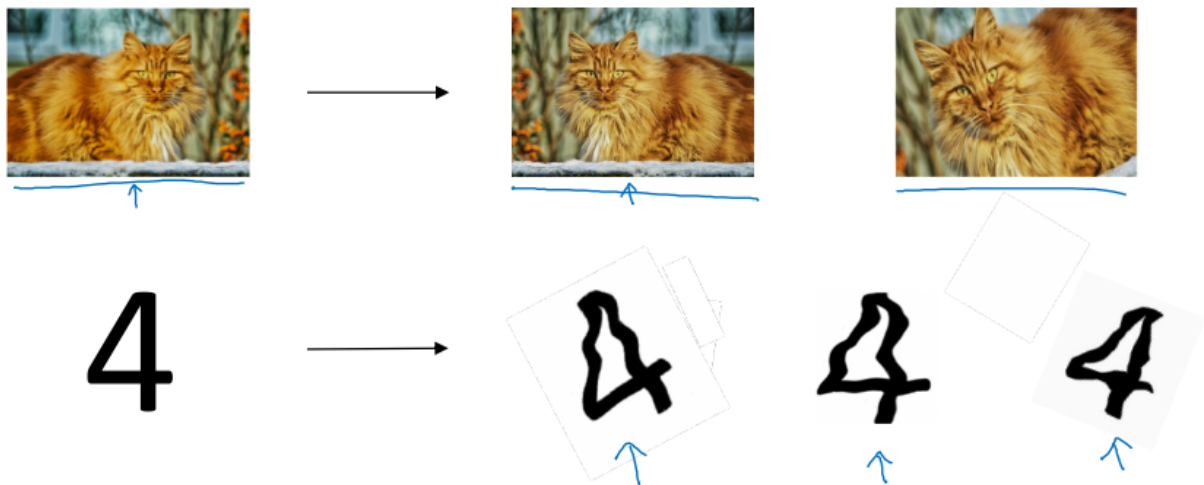
Why does drop-out work?

- 무작위로 유닛이 삭제 되어 특정 feature에 weight이 집중되지 않고 각각의 feature에 가중치가 분산됨 → weights의 제공이 줄어들음
- cv에서는 데이터 수가 항상 부족하기 때문에 dropout 많이 씀
- 단점: 비용함수 J가 잘 정의되지 않음
 - cost function이 감소하는지 확인하기 어려움
 - keep_prob을 1로 설정하고 잘 감소하는지 확인해야함
 - 오버피팅이 일어나지 않는다면 dropout을 사용하지 않는 것이 좋음

다른 정규화 방법들

Data augmentation

- 반전시킨 이미지, crop한 이미지 등 이미지 왜곡과 변형을 통해 데이터 수를 늘림



Early stopping

- dev set error가 최솟값일 때를 최종값으로 선택함.
- 단점
 - optimize J, 오버피팅 방지를 둘 다 함 → 각각을 분리해서 해결하는 것이 더 좋음

