

유런 6주차 변지은

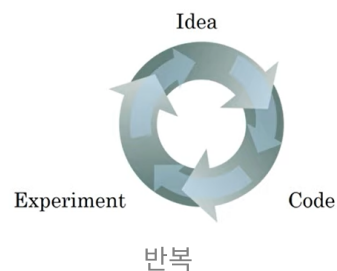
Train/Dev/Test 세트

hyper params

- 신경망이 몇 개의 층 가지는지
- 각 층이 몇 개의 은닉 유닛을 가지는지
- 학습률과 활성화 함수는 무엇인지
- etc

새로운 애플리케이션을 시작할 때는 이 모든 것에 대한 올바른 값을 추측하는 것이 거의 불가능하다.

따라서 실질적으로 더 나은 신경망을 알기 위해선 반복을 통해 알 수 있다.

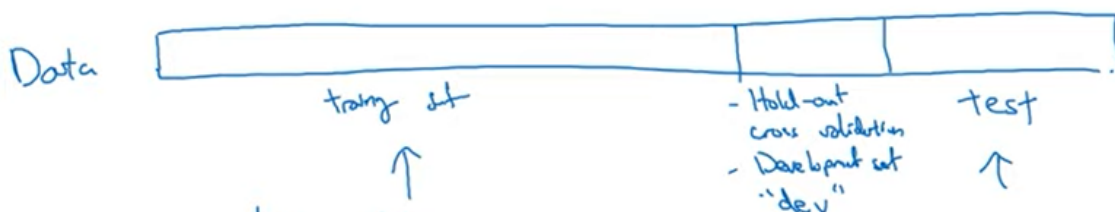


성능 결정 요인

- 가지고 있는 데이터의 양
- 입력 특성의 개수
- 컴퓨터 설정
- 등 다양한 요인에 의해 결정

따라서 빠른 진전을 이루는데 영향을 미치는 것들은 이 사이클을 얼마나 효율적으로 돌 수 있는지와 데이터 세트를 잘 설정하는 것이다.

훈련, 개발(검증), 테스트 세트를 잘 설정하는 것은 과정을 더 효율적으로 만들어주며 반복을 빠르게 해준다.

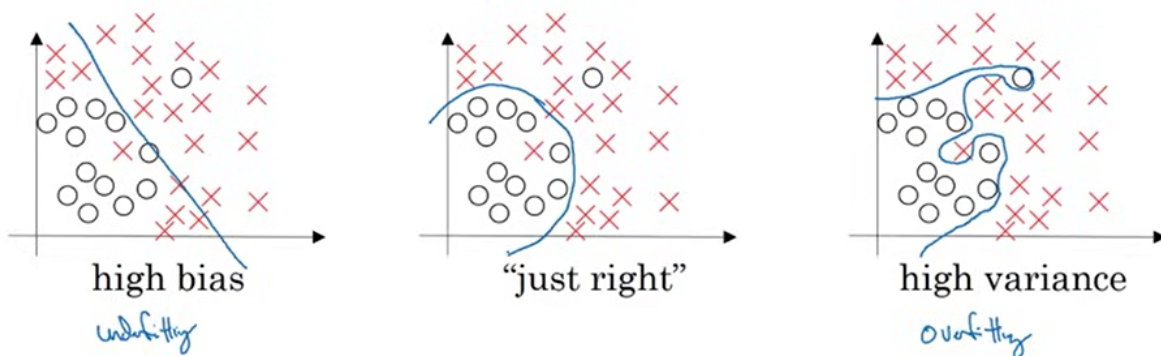


- 훈련 세트 : 훈련을 위해 사용되는 데이터
- 개발 세트 : 다양한 모델 중 어떤 모델이 좋은 성능 나타내는지 확인
- 테스트 세트 : 모델이 얼마나 잘 작동하는지 확인, 테스트 세트의 목표는 최종 네트워크의 성능에 대한 비편향 추정을 제공하는 것이다.

빅데이터 트렌드⇒개발 세트와 테스트 세트가 훨씬 더 작은 비율이 되는 것이 트렌드이다. 왜냐하면 개발 세트는 평가할 수 있을 정도로만 크면 되기 때문이다. 20%, 10% 까지도 필요하지 않다.

편향/분산

Bias and Variance



2차원 데이터의 편향과 분산에 대한 시각화

개 고양이 문제에서

1. Train set error: 1%
2. Dev set error: 11%

라면 훈련 세트에 overfit 되었다고 한다. ⇒ HIGH VARIANCE

1. Train set error: 15%
2. Dev set error: 16%

훈련 데이터에도 잘 맞지 않는다면 데이터에 underfit 된 것이다. ⇒ HIGH BIAS

1. Train set error: 15%

2. Dev set error: 30%

두 개 다 최악.. ⇒ HIGH VARIANCE, HIGH BIAS

1. Train set error: 0.5%

2. Dev set error: 1%

베스트 ⇒ LOW VARIANCE, LOW BIAS

이 모두 베이저안 오차가 다 0에 가깝다고 가정한다(인간은 보통 개 고양이를 다 분류 가능하기 때문이다), 훈련 세트와 개발 세트가 같은 확률 분포에서 왔다고도 가정한다.

Train set 에서 bias 를 확인하고 dev set 에서 variance 를 확인할 수 있다.

머신 러닝을 위한 기본 레시피

기본 레시피

bias 를 확인 하기 위해선 trainig set 를 봐야한다.

⇒ high bias 라면 더 많은 은닉 층 혹은 은닉 유닛을 갖는 네트워크를 선택한다.

⇒ 더 오래 훈련시키거나 다른 발전된 최적화 알고리즘을 사용한다.

⇒ 다양한 신경망 아키텍처를 찾는다(optional.. 위에 것들은 해가 되진 않는다.)

⇒ 데이터를 더 얻는 것은 도움이 되지 않는다.

편향 문제를 해결할 때까지 반복한다.

variance 를 확인 하기 위해선 dev set 를 봐야한다.

⇒ high variance 라면 더 많은 데이터를 얻는게 best

⇒ 정규화 시도

⇒ 다른 신경망 아키텍처를 찾는다

분산 문제를 해결할 때까지 반복한다.

방법이 다르므로 편향과 분산을 잘 파악하는 것이 중요하다.

bias variance trade-off는 이전 머신러닝 연구에서 있었지만 딥러닝 빅데이터 시대로 접어들면서 데이터를 더 많이 갖는 것이 더 큰 네트워크를 훈련시키고 더 많은 데이터를 얻는 것이 더 큰 네트워크를 갖는 것이 분산을 해치지 않고 편향을 감소시킨다. 정규화를 올바르게 했다면!

또한 데이터를 더 얻는 것도 대부분 편향을 해치지 않고 분산을 감소시킨다.

따라서 더 큰 네트워크를 훈련시키거나 더 많은 데이터를 얻는 이 두 단계는 편향만을 감소시키거나 분산만을 감소시키는 틀이 된다. 신경써야 하는 트레이드오프가 확실히 줄어들었다.

정규화

높은 분산을 해결하는 방법 중 하나이다. 데이터를 더 얻는 방법은 비용이 든다.

weight에 대한 정규화 한다. 거의 모든 매개변수는 weight에 있기 때문에 bias는 포함하지 않는다. 원한다면 해도 된다

- 비용함수: $J(w, b) = \frac{1}{m} \sum_{i=1}^m L(y^{(i)}, h^{(i)}) + \frac{\lambda}{2m} \|w\|^2$
- L1 정규화: $\|w\|_1 = \sum_{j=1}^{n_x} |w_j|$
- L2 정규화: $\|w\|_2^2 = \sum_{j=1}^{n_x} w_j^2$

보통 L2 정규화를 훨씬 더 많이 사용한다.

람다는 정규화 매개변수로 불리며 하이퍼파라미터이다. 개발 세트 혹은 교차 검증세트에 주로 쓰인다. 다양한 값을 시도해서 과대적합을 막을 수 있는 최적의 값을 찾는다.

- Frobenius 노름: $\|w^{[l]}\|_F^2 = \sum_{i=1}^{n^{[l-1]}} \sum_{j=1}^{n^{[l]}} (w_{ij}^{[l]})^2$

어떻게 역전파?

$$W^{(2)} := W^{(2)} - \alpha \left[(\text{from backprop}) + \frac{\lambda}{m} W^{(2)} \right]$$

$$= \underbrace{W^{(2)} - \frac{\alpha \lambda}{m} W^{(2)}}_{\text{L2 regularization}} - \alpha (\text{from backprop})$$

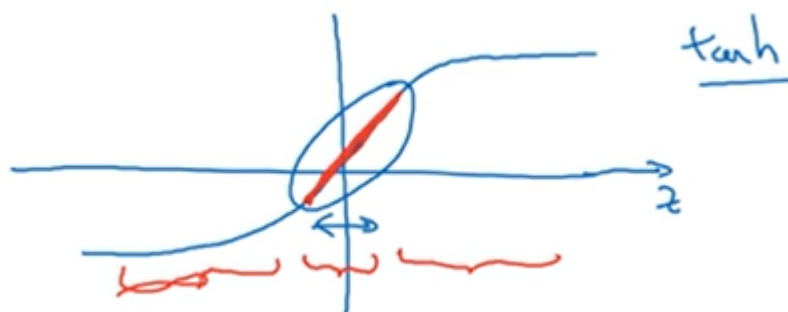
왜 정규화는 과대적합을 줄일 수 있을까요?

가중치 행렬을 막기 위해 보라색 부분이 추가되었었다.

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

람다를 크게 만들어서 가중치 행렬 w 를 0에 상당히 가깝게 설정할 수 있다. 이는 곧 은닉 유닛의 영향력을 죽여준다. 그럼 로지스틱 회귀 유닛에 가까워진다. 그러면 HIGH BIAS 가능성이 생기므로中间的 딱 맞는 경우와 가깝게 하는 적절한 람다 값을 찾는 것이 좋다.드롭아웃 정규화

직관2:



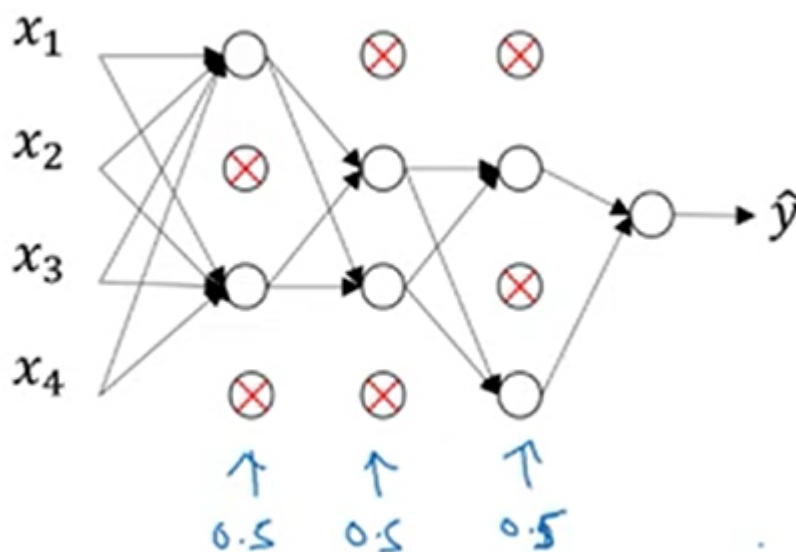
tanh 활성화 함수를 사용했을 경우, 람다 값이 커지면 비용함수에 의해 w 는 작아지고 z 도 작아진다. 이 때 z 구간은 tanh에서 거의 1차원 함수가 되기 때문에 모든 층이 선형 회귀처럼 거의 직선이 된다. \Rightarrow 비선형성을 죽임

드롭아웃 정규화

L2 정규화 외의 정규화 기법

신경망의 각각의 층에 대해 노드를 삭제하는 확률을 설정하여 삭제할 노드를 랜덤으로 선정 후 삭제된 노드의 들어가는 링크와 나가는 링크를 모두 삭제한다.

작아진 네트워크로 다시 훈련 시킨다.



1. 역드롭아웃 : 노드를 삭제 후에 얻은 활성화 값에 삭제하지 않을 확률을 나눠준다. 이로써 기존에 삭제하지 않았을 때 활성화 값의 기대값으로 맞춰질 수 있다.

드롭아웃의 이해

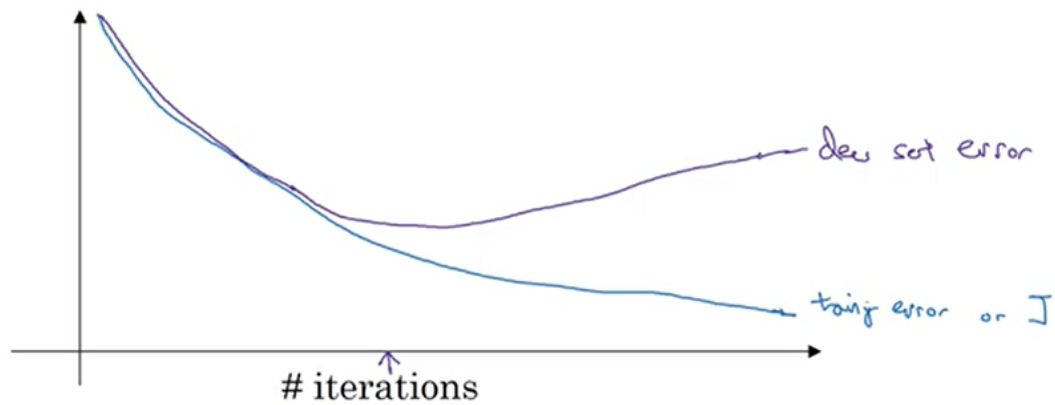
랜덤으로 노드를 삭제시킴으로써 하나의 특성을 의미하는 노드에 의존하지 못하게 함으로써 가중치를 다른 곳으로 분산시킬 수 있다. keep.prop 확률은 층마다 다르게 설정할 수 있다.

드롭 아웃을 입력층에도 쓸 수 있으나 안 쓰는 것이 좋다. 따라서 입력층의 keep.prop 은 1에 가까운 수를 사용한다.

- cv 분야에서는 기본적으로 드롭 아웃 기법을 쓴다.
- 드롭 아웃을 할 때는 경사 하강이 잘 되고 있는지를 확인해야 한다

다른 정규화 방법들

- 데이터 증식
 - 이미지를 증강시켜 데이터를 늘릴 수 있다.(반전, 회전, 변형, 왜곡)
 - 컴퓨터적 비용이 안 든다
- 조기 종료
 - 앞쪽 w 는 0에 가깝고 반복을 실행할수록 w 는 커진다. 중간에 멈추면 w 가 중간 크기의 값을 갖는 상태이다. 따라서 신경망이 덜 과대적합된다.
 - 단점: 비용함수를 최적화하는 것도 멈추게 된다



조기 종료는 신경망이 이 반복 주변에서
가장 잘 작동하는 것을 알 수 있습니다

Andrew N