[딥러닝 2단계] 1. 머신러닝 어플리케이션 설정하기

1. Train/Dev/Test 세트

Applied ML is a highly iterative process

- 새로운 애플리케이션을 시작할 때는 모든 것에 대한 올바른 값을 추측하는 것이 거의 불가능
- 실질적으로 머신러닝을 적용하는 것은 매우 반복적인 과정
- 사이클 반복을 통해 좋은 최적의 하이퍼 파라미터 값을 찾아야 한다
- 빠른 진전을 위해 1) 사이클 얼마나 효율적으로 돌 수 있는가 2) 테스트 세트 설정이 중요

Train/dev/test sets

- 훈련, 개발, 테스트 세트를 잘 설정하는 것을 과정을 효율적으로 만든다

전통적인 방법

- 데이터 일부를 잘라 `훈련 세트`, 다른 일부는 `교차 검증 세트(개발(dev) 세트)`, 마지막 부분은 `테스트 세트`로 만든다
    1. 훈련세트 : 훈련을 위해 사용되는 데이터
    2. 개발 세트 : 다양한 모델 중 어떤 모델이 가장 좋은 성능을 내는지 확인
    3. 테스트 세트 : 더 발전하고 싶은 최종 모델이 나오면 알고리즘이 얼마나 잘 작동하는지 편향 없이 측정
- 70(훈련 세트)+30(테스트 세트) or 60(테스트 세트)+20(개발 세트)+20(테스트 세트)로 나누는게 일반적인 관행
    - 몇년전까지 이것은 머신러닝에서 경험에서 나온 최적의 관행으로 여겨졌다
    - 상대적으로 적은 데이터 세트일 경우에는 전통적인 비율로 설정하는 것도 괜찮음
- 빅데이터 시대에는 개발 세트와 테스트 세트가 훨씬 더 작은 비율이 되는 것이 트렌드
    - 개발 세트

        - 개발 세트는 서로 다른 알고리즘을 시험하고 어떤 알고리즘이 더 잘 작동하는지 확인하는 것
        - 개발 세트는 평가할 수 있을 정도로만 크면 되므로 전체 데이터의 20%나 필요하지 X
    - 테스트 세트
        - 테스트 세트는 최종 알고리즘이 어느 정도 성능인지 신뢰 있는 추정치 제공
        - 100만개의 샘플이 있으면 만 개의 샘플을 설정해도 충분

Mismatched train/test distribution

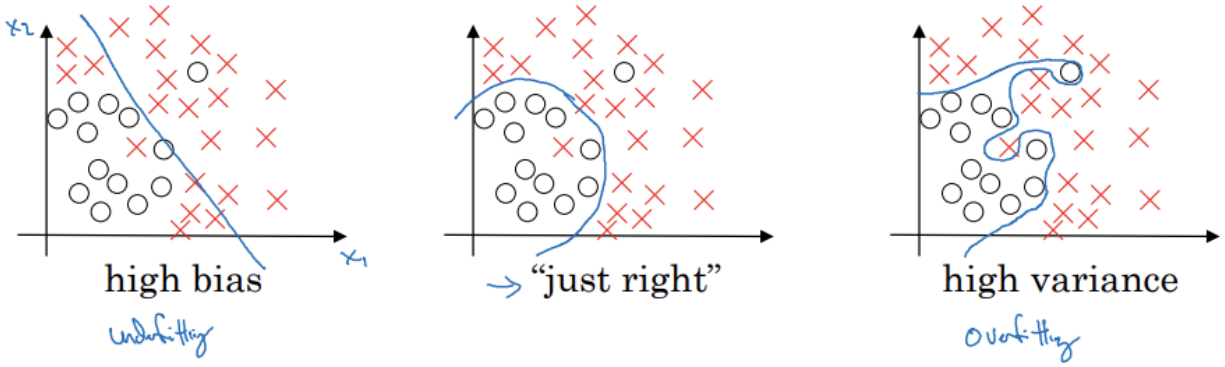
- 일치하지 않는 훈련/테스트 분포에서 훈련
- 훈련 세트 : 인터넷에서 다운 받은 고양이 사진
- 개발/테스트 세트 : 앱을 사용하는 사용자들이 업로드한 사진
  -> 두 가지 데이터의 분포는 달라질 수 있음
- 개발/테스트 분포가 같은 분포에서 오는 게 좋음
- 대량의 훈련 데이터가 필요하기 때문에 훈련 세트는 개발/테스트 세트와 다른 분포일 수 있음
- 테스트 세트를 갖지 않아도 괜찮음
    - 비편향 추정이 필요 없는 경우에 테스트 세트 없어도 괜찮음

개발 세트만 있는 경우 (train/dev(test) set)

- 모든 테스트 세트를 훈련 세트에서 훈련시키고 다른 모델 아키텍트를 시도
- 이것을 개발세트에서 평가
- train/test 세트라고 부르지만 실제로는 테스트 세트를 교차 검증 세트로 사용
    - 좋은 용어 x -> 테스트 세트에 과적합이 일어날 수 있음
    - train/dev set이 더 올바른 용어

2. 편향/분산

Bias and Variance



높은 편향 (high bias)	알맞음 (just right)	높은 분산 (high variance)
과소적합 (underfitting)		과대적합 (overfitting)

훈련 세트와 개발 세트의 관계

가정)

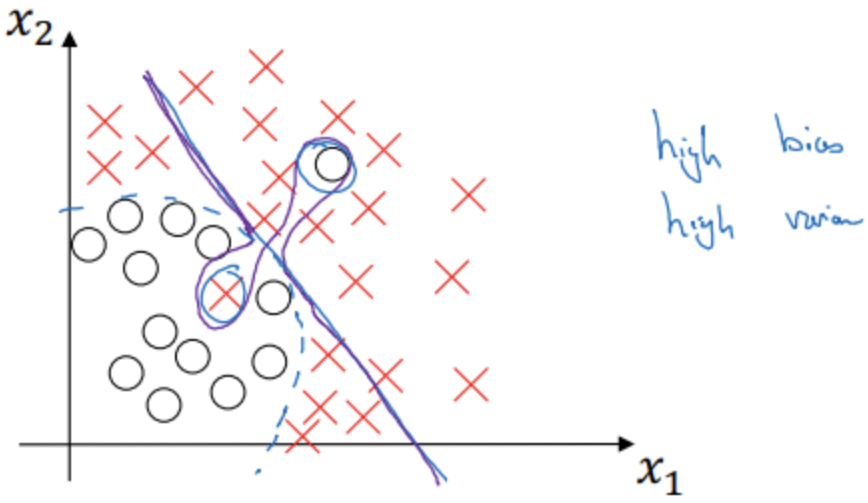
1. 인간 수준의 성능 0% -> 베이지안 최적 오차가 0%
2. 훈련 세트와 개발 세트가 같은 확률 분포

	1. high variance	2. high bias	3. high bias, high variance	4. low bias, low variance
Train set error	1%	15%	15%	0.5%
Dev set error	11%	16%	30%	1%

1. high variance : 훈련 세트에 과대적합 되어 개발 세트가 있는 교차 검증 세트에서 일반화 되지 못한 경우
2. high bias : 훈련 세트에 대해서도 잘 맞지 않음 = 데이터에 과소적합, 개발세트에서는 합리적인 수준의 일반화
3. high bias, high variance : 훈련 세트에 잘 맞지 않음, 개발 세트에서는 훨씬 더 나쁜 오차
4. low bias, low variance : 오차가 낮음

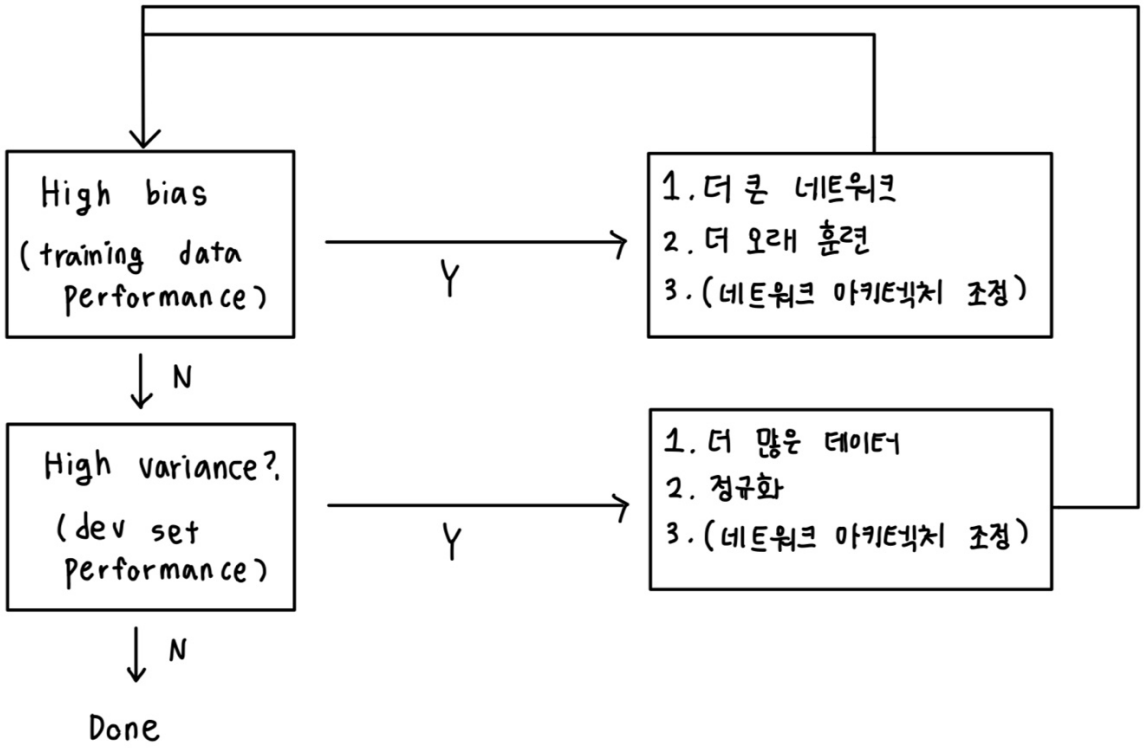
💡 Train set error : 훈련 데이터에서 알고리즘이 얼마나 적합한지, 즉 편향이 얼마나 높은지 알 수 있음
Dev set error : 훈련 세트에서 개발 세트로 갈 때 오차가 얼마나 커지는지에 따라서 분산이 얼마나 높은지 알 수 있음

High bias and high variance



- 이 분류기는 거의 선형이지만 곡선이나 이차함수가 필요하기 때문에 높은 편향
- 중간에 잘못 라벨링된 샘플을 맞추기 위해 너무 많은 굴곡을 가지게 되어 높은 분산

3. 머신러닝을 위한 기본 레시피



High bias

- 훈련 세트 혹은 훈련 데이터 성능을 봐야한다

해결 방법

- 더 큰 네트워크
  - 더 많은 은닉층, 은닉 유닛을 갖는 네트워크 선택
- 더 오래 훈련
- (네트워크 아키텍처 조정)
  - 작동 안할 수도 있음

High variance

- 개발 세트 성능을 일반화 할 수 있는가

해결 방법

- 더 많은 데이터 구하기

    - 얻을 수 있다면 가장 좋은 방법
- 정규화
- (네트워크 아키텍처 조정)
    - 완전히 체계화하기는 어렵

중요한 점

1. 높은 편향이냐 분산이냐에 따라 시도하는 방법 매우 다르다
    - 훈련이나 개발 세트를 편향이나 분산 문제가 있는지 진단하는데 사용하고 방법 선택
2. 초기 머신러닝 시대에는 편향-분산 트레이드오프에 대한 많은 논의가 있었다
    - 시도할 수 있는 많은 것들이 편향 증가/분산 감소 or 편향 감소/분산 증가 시켰기 때문
    - 현재 딥러닝 빅데이터 시대에는 1) 더 큰 네트워크, 2) 더 많은 데이터는 편향만을 감소시키거나 분산만을 감소시키는 툴이라는 것을 알게되었다
    - 지도학습은 트레이드오프가 훨씬 적기 때문에 딥러닝에서 유용