1. 머신러닝 어플리케이션 설정

Summary

Train/Dev/Test set

- 하이퍼파라미터 튜닝의 반복적인 과정을 빠르게 하기 위해 효율적으로 trian/dev/test set을 분리해야 함
- 전통적인 비율: 7/3 or 6/2/2
- 빅데이터: 98/1/1
- dev, test set은 같은 분포에서 와야함

bias / variance

- training의 성능이 낮은 경우 → high bias
- test의 성능이 낮은 경우 → high variance

알고리즘 학습 방법

1. bias가 높을 때
    a. try bigger network (신경망 깊이, 은닉층 수 조정)
    b. train longer
    c. try different NN architecture

1. variance가 높을 때
    a. more data
    b. regularization
    c. try different NN architecture

- bias를 낮추는 방법이 variance도 낮추지는 않음 (그 반대도 마찬가지)
- bias variance tradeoff
  - bias를 낮추는 것이 variance를 높임. 또는 그 반대
  - 최근에는 bias variance tradeoff의 영향이 거의 없음

Train/Dev/Test set

효율적인 train/dev/test set 분리가 필요한 이유

- 하이퍼 파라미터를 처음부터 최적화하는 건 불가능함
  - hyper parameters: #layers, #hidden units, learning rates, activation functions
- 최적화 하이퍼 파라미터를 찾는 **반복적인 과정** 필요
- 반복 과정을 빠르게 할 수 있도록 train/dev/test set을 효율적으로 나눠야 함

split data

- train set + development set + test set
- 데이터가 적을 경우에는 7:3 또는 6:2:2
- 빅데이터 시대에서는 dev, test 비율을 줄임
  - dev, test에는 많은 양의 데이터가 필요하지 않기 때문
  - ex. 98:1:1

mismatched train/test distribution

- dev, test set은 같은 분포여야 함
- ex.
  - training set: cat pictures from webpages
  - dev, test set: cat pictures from users using the app
- 비편향 추정을 하지 않는다면 test set은 없어도 괜찮음 (only train/dev)

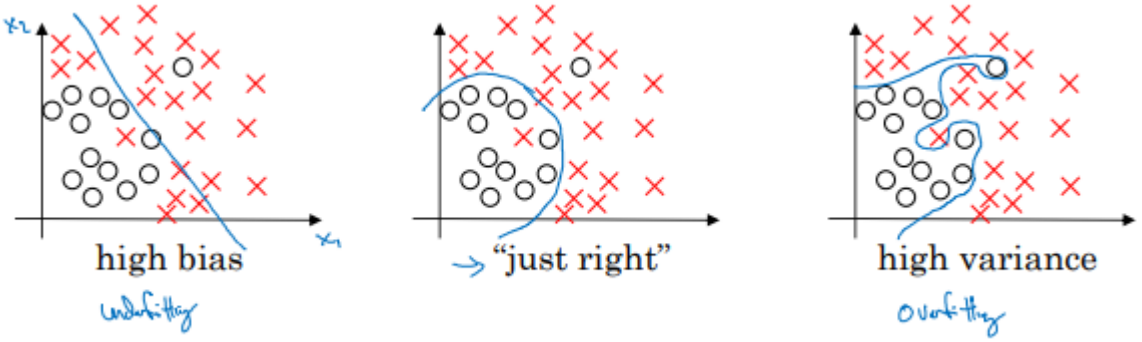
※ 비편향 추정 (unbiased estimate)

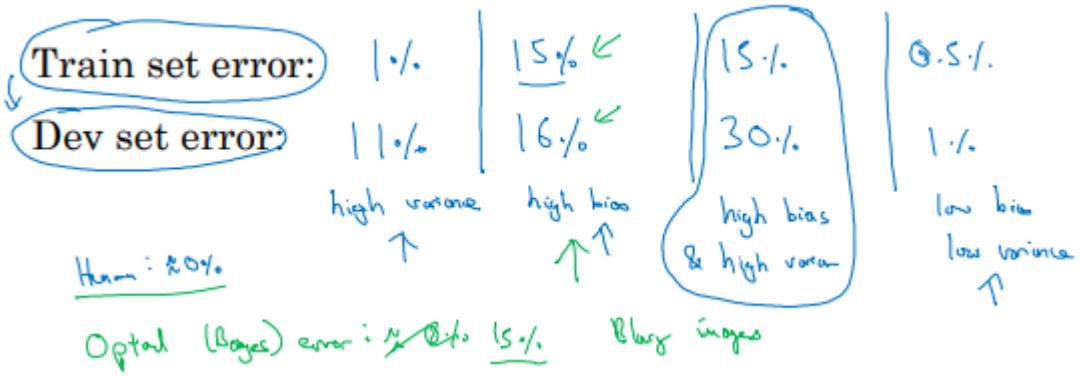
- bias: 추정량의 기댓값과 실제 파라미터 값 간의 차이
- unbiased: 기댓값 = 실제 파라미터 값
  - 무작위로 표본을 뽑아 추정했을 때, 평균적으로 실제값과 일치함

편향/분산

bias / variance

- **bias:** 모델의 예측값과 실제값의 차이
  - training의 성능이 높으면 low bias
  - low bias = 모델이 너무 간단함 (underfitting)
- **variance:** 모델의 예측이 주변 데이터 포인트에 대해 얼마나 퍼져있는지
  - test의 성능이 높으면 low variance
  - low variance = 모델이 너무 복잡함 (overfitting)





train set error은 작지만 dev set error가 높은 경우

- overfitting
- high variance

train set, dev set error가 비슷하게 높은 경우

- underfitting
- high bias

train set error가 크고 dev set error가 더 큰 경우

- high bias
- high variance

둘 다 작은 경우

- low bias
- low variance

가정: 베이즈 오차가 작고 train,dev set이 같은 분포에서 왔음

Basic Recipe for ML

알고리즘 평가 방법

- high bias?
  - train set의 성능을 보고 평가
  - try:

    - bigger network(more hidden layers)
    - train longer
    - different NN architectre
- high variance?
    - dev set performance로 평가
    - try:
        - more data
        - regularization
        - differnet NN architecture

bias variance tradeoff

- 모델의 복잡성을 높이는 것은 bias를 감소시키고 variance는 증가시킴
- 최근에는 정규화 등의 영향으로 bias variance tradeoff 문제가 많이 해결됨