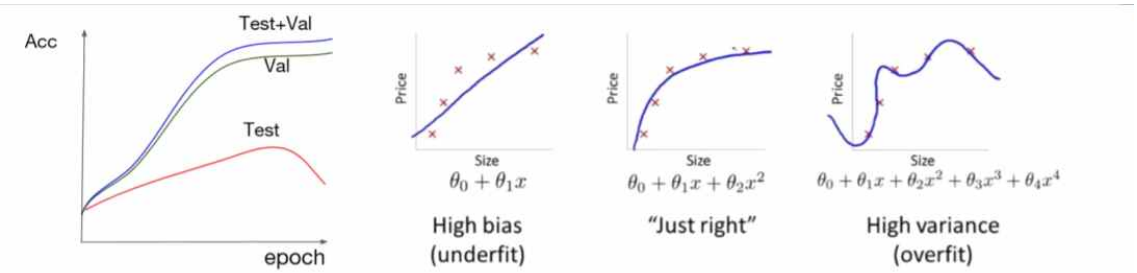
과제명	머신러닝 9주차 과제

- 과 목 명 머신러닝
- 담당교수 김 용 운
- 학부(과) 컴퓨터소프트웨어공학과
- 학 년 4
- 분반번호 01
- 학 번 20173147
- 이 름 이 명 진
- 연 락 처 010-2999-7748
- 제 출 일 2020 년 05 월 18 일

09-01 Application And Tip (2)

지난 시간,
어플리케이션의 팁 : learning rate, 데이터 전처리



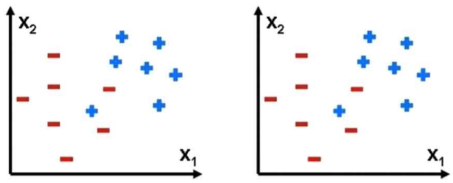
이번 시간 주제: Overfitting -> 머신러닝에서 가장 큰 문제점.
의미 해석: over->과하게 , fitting -> 맞춰져 있다.

학습이 반복되면 되면 평가를 진행하면 녹색선과 같이 모델의 정확도가 증가하게됨.
그러나, 데이터가 적기 때문에 그 데이터에 맞게 실제 모델이 만들어진 경우.
물론 평가나 테스트를 한다고 하면 그렇지 않음. 즉, 모델에서 사용하지 않은
데이터가 들어오게되면 빨간 선과 같이 정확도가 떨어지게됨.
학습을 많이 하게되면 -> 정확도가 증가할 거야 라는 생각을 갖게됨.

bias(무언가 편향되어있다) 그래프를 살펴보면 학습이 이상적으로 됐다고 보는 건
just right, 학습이 덜 이루어진 것은 underfit, 학습이 이상적으로 되지 않음 ->
overfit(variance 학습을 많이해서 변화량이 높은 것).

학습을 많이하게되면 -> 학습한 데이터만 인식을 하게되고 새로운 데이터가 들어왔을
때 학습을 하지 못함. 이를 overfitting이라고 한다.

어떠한 모델을 학습한다고 했을 때 가장 이상적인 모델은 무엇인가?



기울기가 -1인 직선을 x2와 x1 점을 기준으로 그려보면 이상적인 모델이라고 볼 수
있음. 그러나 과도하게 학습을 하다보면 + 데이터 하나를 인식하지 못함.

이를 해결하기 위한 방법은 다음과 같다.

• Overfitting을 해결하기 위한 방법

✓Set a features

» Get more training data -> 더 많은 데이터를 집어넣으면 overfitting 방지 가능.

» Smaller set of features -> feature들의 개수를 줄이자(차수를 줄이면 인식률 증가)

» Add additional features -> 오히려 feature의 개수를 더 늘리자(많아지면 많아질수록 학습이 덜 되었을 때 즉, 의미있는 데이터가 별로 없을 때) 물론, 어느 특정의 의미있는 부분을 찾은 이후에는 2번째 방법과 같이 에러율이 증가한다.

✓Regularization (Add term to loss) -> 람다 값들은 더 줘서 하는 방법

추후에 Neural Network에 대해 배운 이후에 설명 예정.

✓Dropout(0.5 is common)

✓Batch Normalization

Overfitting 너무 학습이 많아져 변화량이 많으니 내가 학습한 모델이 대해서만 정확도가 높은 것이지 새로운 데이터가 들어오면 학습이 힘든 상황 -> 학습이 잘 되는 적정 수준을 찾아야함.

==========

09-02 Application And Tip (3)

overfiting -> 머신러닝에서 가장 골치 아픈 문제.
학습을 할 때 테스트 데이터 셋들은 어떻게 구성해야하는지와 어떻게 했을 때 학습이 가장 잘 되는 것인지에 대해 알아보자.

• Performance evaluation: is this good?

data -> MODEL -> 우리가 예측하는 가설값이 나오게 됨.
학습을 할 때 트레이닝 셋은 어떻게 구성해야하는가?

Size	Price
2104	400
1600	330
2400	369
1416	232
3000	540
1985	300
1534	315
1427	199
1380	121
1494	243

매칭을 시켜서 컴퓨터보고 외우라고 하면 ex) if (x== 2014)

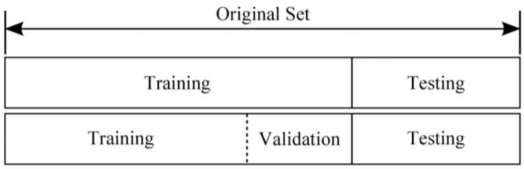
y= 400

처럼 트레이닝 셋을 구성을 하면 과연 좋을까? => NO!!

데이터 셋 중에서 트레이닝 셋 70%, 테스트 셋 30%으로 이용.
테스트 셋은 꽁꽁 숨겨놓는다. 이후에 학습을 시킨다. 이후에 예측값과 테스트 셋의 데이터들을 비교했을 때 인식률이 좋게 나타나면 학습이 잘 된 것.
즉, 처음부터 전체를 이용하는 것이 아님.

다른 경우, validation을 해야하는 경우.

• Training, validation and test sets



트레이닝 셋 70%, 테스트 셋 30%
위와 같이 구성되어있는 데이터들을 가지고 학습을 진행하면 된다.
데이터는 많으면 많을수록 좋다.

online learning -> 데이터가 천 만개 있다고 했을 때 모델이 학습을 하는데에 속도가 엄청나게 느릴 것은 뻔함. 온라인 러닝을 사용하면 천 만개의 데이터를 쪼개서 10만개 정도를 먼저 학습하고 마무리가 되면 그 다음 10만개를 넣어서 학습. 물론 이전에 학습된 데이터들은 온전히 저장. 천 만개를 한 번에 돌리는 것보다 속도가 훨씬 빠름. 또 다른 장점은 10만개의 데이터가 추후에 100만개로 늘었다면 이전에 저장되있는 10만개 때문에 90만개만 학습을 시키면됨 -> 인식률 또한 증가. 정적이 아닌 동적으로 활용. -> ex) 구글의 한국어 인식률, 과거보다 크게 인식률 향상. 처음에 만들 때는 물론 오프라인으로 할 수 밖에 없음. 어느정도 학습이 되었다고 생각되면 온라인 러닝으로 변경.

비교해보면 다음과 같다.

• Online learning vs Batch learning

	Online Learning	Batch(Offline) Learning
Data	Fresh	Static
Network	connected	disconnected
Model	Updating	Static
Weight	Tunning	initialize
Infra(GPU)	Always	Per call
Application	Realtime Process	Stopping
Priority	Speed	Correctness

온라인 러닝의 가장 우선순위는 속도! 배치는 초기에 먼저 이루어짐.

==========

09-03 Lab_Application And Tip (1)

실습!