DL 스터디&프로젝트

[Euron 중급 세션 5주차] 딥러닝 2단계 1. 머신러닝 어플리케이션 설정하기

by 공부하자_ 2023. 10. 9.

딥러닝 2단계: 심층 신경망 성능 향상시키기

1. 머신러닝 어플리케이션 설정하기

📌 Train/Dev/Test 세트(C2W1L01)

핵심어: 훈련 세트(Train set), 개발 세트(dev set), 테스트 세트(test set)

지금까지는 신경망을 어떻게 구현하는지 학습했다면, 이제부터는 신경망이 잘 작도하기 위한 실질적인 면을 배울 것. 데이터를 설정하기 위한 하이퍼파라미터 튜닝부터 최적화 알고리즘의 속도를 높여 적당한 시간 안에 학습 알고리즘이 학습할 수 있도록 하는 방법까지 다뤄볼 예정.

이번 시간에는 머신러닝 문제를 어떻게 해결하는지에 관해 이야기할 것. 정규화에 대해 다루고 신경망 구현이 맞게 되었는지 확인하는 몇 가지 기술을 살펴보고자 함.

분류 전체보기

DL 스터디&프로젝트

Data Science 프로젝트

Github 스터디

Data Science 개인 공부

Backend 프로젝트

기타 공부

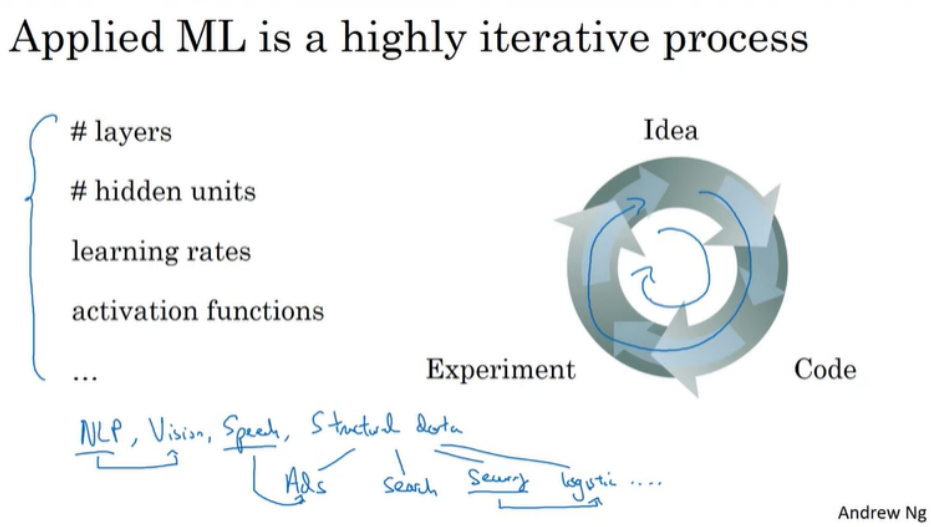
공지사항

최근글 인기글

[Euron 중급 세션 5주차] ...
2023.10.09

백엔드 프로젝트 5주차 (SQL 첫걸음) - 5장
2023.10.07

백엔드 프로젝트 4주차 (SQL 첫걸음) - 4장



훈련, 개발, 테스트 세트를 어떻게 설정할지에 관한 좋은 선택을 내리는 것은 좋은 성능을 내는 네트워크를 빠르게 찾는데 큰 영향을 줌. 신경망을 훈련시킬 때는 많은 결정을 내려야 하는데, 신경망이 몇 개의 층을 가지는지, 각각의 층이 몇 개의 은닉 유닛을 가지는지, 학습률이 무엇인지, 서로 다른 층에 사용하는 활성화 함수는 무엇인지 등이 있음. 새로운 애플리케이션을 시작할때는 이 모든 것에 대한 올바른 값을 추측하는 것이 거의 불가능함. 다른 하이퍼파라미터에 대해서도 마찬가지임. 따라서 실질적으로 머신러닝을 적용하는 것은 매우 반복적인 과정. 주로 처음에는 아이디어로 시작하여 특정 개수의 층과 유닛을 가지고 특정 데이터 세트에 맞는 신경망을 만듦. 그럼 이것을 코드로 작성하고 실행하고 실험을 진행함. 그 결과 특정 네트워크 혹은 설정이 얼마나 잘 작동하는지를 알게 되고, 이 결과에 기반해 아이디어를 개선하고 몇 가지 선택을 바꾸게 됨. 그리고 더 나은 신경망을 찾기 위해 이 과정을 반복함.

오늘날 딥러닝은 자연어 처리 과정부터 컴퓨터 비전, 음성 인식, 구조화된 데이터에 적용된 다양한 애플리케이션(광고, 웹 서치, 컴퓨터 보안, 물건 배송 등)과 같이 여러 분야에서 발전을 이뤄옴. 가끔 NLP에 많은 경험을 가진 연구원이 컴퓨터 비전에서 무언가를 시도하거나, 음성 인식에 많은 경험을 가진 연구원이 광고에서 무언가를 시도하거나 하는 일들이 있으나, 일반적으로 어떤 분야나 애플리케이션의 직관이 다른 애플리케이션 영역에 거의 적용되지 않음. 그리고 최고의 선택은 가지고 있는 데이터의 양, 입력 특성의 개수, GPU나 CPU처럼 훈련을 진행하는 컴퓨터 걸정, 정확이 어떤 설정을 했는지 등 다양한 요인에 의해 결정됨. 따라서 많은 애플리케이션에서 딥러닝에 아주 경험이 많은 사람일지라 첫 시도에 하이퍼파라미터에 대한 최고의 선택을 하는 것은 거의 불가능함. 이와 같이 오늘날 딥러닝을 적용하는 것은 애플리케이션의 네트워크에 대한 좋은 선택을 찾기 위해 이 사이클을 여러 번 돌아야하는 매우 반복적인 과정임. 빠른 진전을 이루는 데 영향을 미치는 것들은 이 사이클을 얼마나 효율적으로 돌 수 있는지와 데이터 세트를 잘 설정하는 것임.

2023.10.07

[Euron 중급 세션 4주차] ...
2023.10.02

백엔드 프로젝트 4주차 (SQL 첫걸음) - 3장
2023.09.30

최근댓글

좋은 글 잘 보고 가요! 감사...

좋은 글 잘 보고 가요! 감사...

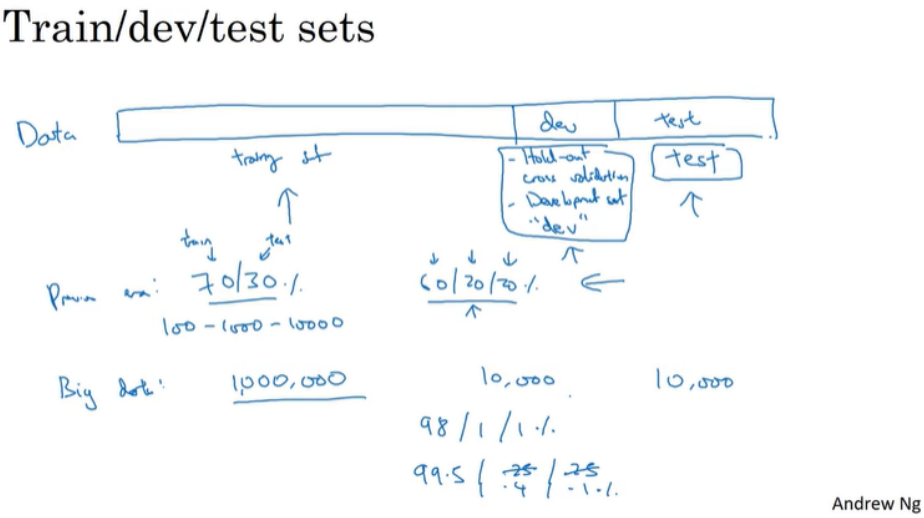
잘보고 갑니다!

포스팅 잘보고 갑니다! 응원...

태그

데이터분석, 이지스퍼블리싱, Doit, pandas, 판다스, bda, 딥러닝스터디, 딥러닝교과서, 데이터사이언스, 판다스입문

전체 방문자
483
Today : 0
Yesterday : 2

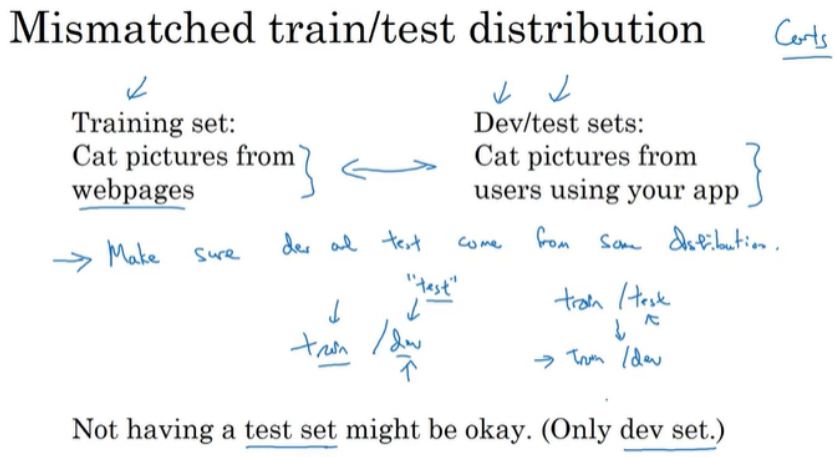


Andrew Ng

훈련, 개발, 테스트 세트를 잘 설정하는 것은 과정을 더욱 효율적으로 만들어 줌. 여기 훈련 데이터를 큰 박스로 나타내면, 전통적인 방법은 모든 데이터를 가져오고 일부를 잘라서 훈련 세트를 만든 후, 다른 일부는 교차 검증 세트(개발 세트), 마지막 부분은 테스트 세트로 만들게 됨. 작업의 흐름은 훈련 세트에 계속 훈련 알고리즘을 적용시켜 개발 세트 혹은 교차 검증 세트에 대해 다양한 모델 중 어떤 모델이 가장 좋은 성능을 내는지 확인함. 이 과정을 충분히 거치고 더 발전시키고 싶은 최종 모델이 나오면 테스트 세트에 그 모델을 적용시켜 알고리즘이 얼마나 잘 작동하는지 편향 없이 측정하게 됨. 머신러닝 이전의 시대에서는 모든 데이터를 가져와서 훈련세트 70대 테스트세트 30으로 나누는 것이 일반적인 관행이었으며(명시적인 개발 세트가 없는 경우), 혹은 훈련세트 60, 개발세트 20, 테스트세트 20으로 나누기도 했음. 몇 년 전까지만 해도 이는 머신러닝에서 최적의 관행으로 여겨졌는데, 총 100개, 1000개, 10000개의 샘플의 경우 이 비율은 경험에서 나온 가장 합당한 비율이었음.

그러나 총 100만 개 이상의 샘플이 있는 현대 빅데이터 시대에는 개발 세트와 테스트 세트가 훨씬 더 작은 비율이 되는 것이 트렌드가 되었음. 그 이유는 개발 세트와 테스트 세트의 목표는 서로 다른 알고리즘을 시험하고 어떤 알고리즘이 더 잘 작동하는지 확인하는 것이기 때문에 개발 세트는 평가할 수 있을 정도로만 크면 됨(2개, 10개의 알고리즘 선택 중 어느 것이 더 나은지 빠르게 선택할 수 있도록). 이를 위해 전체 데이터의 20퍼센트나 필요하지는 않음. 따라서 예를 들어 백만 개의 훈련 샘플이 있는 경우, 만 개의 샘플을 개발 세트로 설정하면 두 개의 알고리즘 중 어느 것이 더 좋은지 평가하기에 충분함. 같은 방식으로, 테스트 세트의 주요 목표는 최종 분류기가 어느 정도 성능인지에 대해 신뢰할만한 추정치를 제공하는 것이므로 마찬가지로 백만 개의 샘플이 있다면 만 개의 샘플을 설정해도 충분함. 따라서 백만 개의 샘플을 가지고 있는 경우, 개발 세트로 만 개, 테스트 세트로 만 개 설정하면 만 개는 백만 개의 1%이므로 98% 훈련 세트, 1%의 개발과 테스트 세트로 나뉨. 백만 개보다 많은 샘플을 가지는 애플리케이션의 경우, 99.5%의 훈련 세트, 0.25%의 개발 세트, 테스트 세트가 될 것. 다시 말하자면 머신 러닝 문제를 설정할 때는 훈련, 개발, 테스트 세트를 설정하는데 상대적으로 적은 데이터 세트의 경우 전통적인 비율로 설정하는 것도

괜찮음. 그러나 훨씬 더 큰 데이터 세트라면 개발과 테스트 세트를 전체의 20% 혹은 10%보다 더 작게 설정하는 것도 괜찮은 방법임.



Andrew Ng

현대 딥러닝의 또 다른 트렌드는 더 많은 사람들이 일치하지 않는 훈련, 테스트 분포에서 훈련시킨다는 것. 사용자가 모든 사진들을 업로드하는 앱을 만든다고 가정하고, 사용자에게 그 중 고양이 사진을 찾아 보여주는 것을 목표로 설정. 아마 훈련 세트는 인터넷에서 다운 받은 고양이 사진이 될 것. 그러나 개발과 테스트 세트는 앱을 사용하는 사용자들에 의해 구성된 것임. 따라서 훈련 세트는 인터넷에서 긁어온 많은 사진이고, 개발과 테스트 세트는 사용자에 의해 업로드된 사진들임. 많은 웹페이지는 아주 전문가스럽고 잘 정돈된 고양이 사진 결과를 가지고 있으나 사용자들은 흐릿한 저해상도의 사진들을 올릴 것. 따라서 두 가지 데이터의 분포는 달라질 수 있음. 경험상 이런 경우 개발과 테스트 세트가 같은 분포에서 와야 함. 개발 세트를 사용해 다양한 모델을 평가하고 성능을 개선하기 위해 열심히 노력할 것이기 때문. 그러나 딥러닝 알고리즘은 대량의 훈련 데이터가 필요하기 때문에, 더 많은 훈련 세트를 모으기 위해 웹 페이지를 크롤링하는 등 모든 창의적인 전략을 사용하게 될 것(개발이나 테스트 세트와 같은 분포에서 훈련 세트 데이터가 오지 않는 경우임에도 불구하고). 이러한 경험적인 규칙을 따를수록 머신러닝 알고리즘의 진전은 더 빠르게 일어날 것.

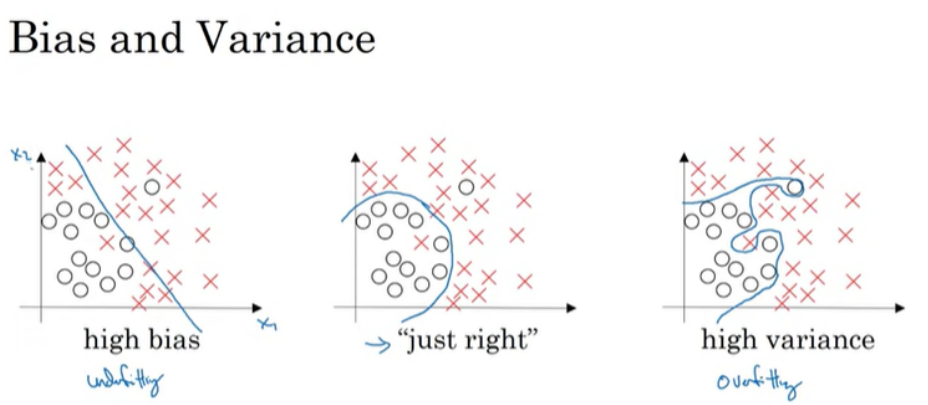
마지막으로, 테스트 세트를 갖지 않아도 괜찮음. 테스트 세트의 목표는 최종 네트워크의 성능에 대한 비편향 추정을 제공하는 것인데, 비편향 추정이 필요 없는 경우 테스트 세트를 갖지 않아도 괜찮음. 따라서 개발 세트만 있는 경우 모든 테스트 세트를 훈련 세트에서 훈련시키고 다른 모델 아키텍트를 시도하고 이것을 개발 세트에 평가함. 그리고 이 과정을 반복해 좋은 모델을 찾음. 이는 개발 세트에 데이터를 맞추기 때문에 성능에 대한 비편향 추정을 주지 않으며, 따라서 비편향 추정이 필요하지 않다면 테스트 세트가 없어도 괜찮음. 머신러닝에서 별도의 테스트 세트 없이 훈련 세트와 개발 세트만 있는 경우 대부분의 사람들은 개발 세트를 테스트 세트라고 부름. 그러나 실제로 하는 것은 테스트 세트를 교차 검증 세트로 활용하는 것. 다만 이 경우 테스트 세트에 과적합하기 때문에 완벽히 좋은 용어라고는 할 수 없음. 따라서 훈련 세트와 테스트 세트만 있다고 말하는 경우, 그들이 훈련과 개발 세트를 진짜 가지고 있

는지 주의 깊게 봐야 할 필요가 있음. 왜냐하면 테스트 세트에 과적합이 일어날 수 있기 때문. 학습/개발 세트로 부르는 것이 더 올바른 용어라고 생각하더라도, 문화적으로 용어를 바꿔 학습/테스트 세트 대신 학습/개발 세트로 부르는 것이 어려울 수 있음. 하지만 알고리즘의 성능에 완전한 비편향 추정이 필요 없다면 이러한 방식도 괜찮음.

📌 편향/분산(C2W1L02)

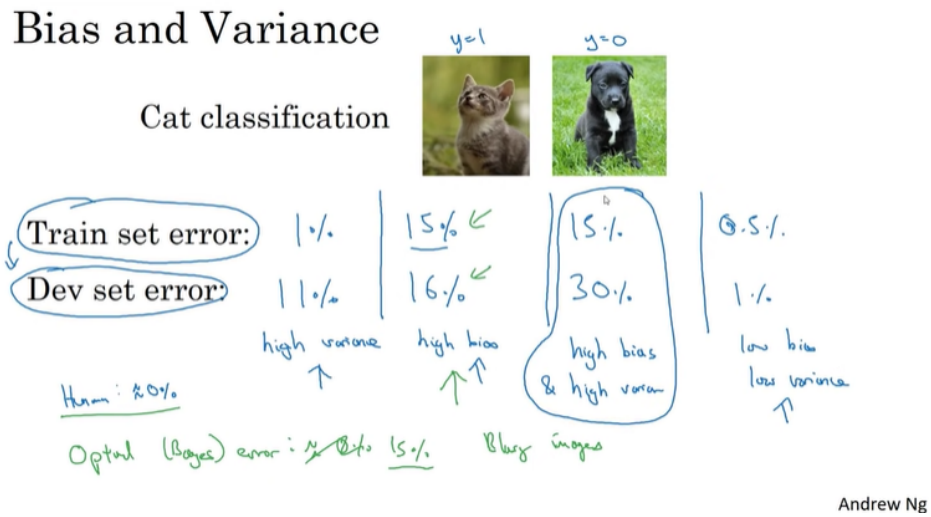
핵심어: 편향(Bias), 분산(Variance), 편향-분산 트레이드오프(Bias-Variance trade-off)

대부분의 유능한 머신러닝 실무자들은 편향과 분산에 대한 수준 높은 이해를 가지고 있는 경우가 많음. 편향과 분산에 대한 개념은 배우기 쉽지만 예상보다 미묘하기 때문에 완벽한 학습은 어려울 수 있음. 딥러닝 시대의 또 다른 트렌드로는 편향-분산 트레이드오프에 관한 더 적은 논의인데, 딥러닝 시대에 여전히 편향과 분산 이야기는 하지만 편향-분산 트레이드오프에 대한 이야기는 줄어들었음.



이것이 무슨 말인지 더 알아보자면, 위와 같이 데이터들이 있고 각 데이터에 맞게 직선을 넣는다고 할 때, 로지스틱 회귀라고 가정하면 데이터에 잘 맞는 형태는 아님. 높은 편향값의 클래스이므로 데이터의 과소적합이라고 말함. 반대로 아주 복잡한 분류기를 사용하면(깊은 신경망, 많은 은닉 유닛 등) 신경망을 사용하는 경우, 데이터를 완벽하게 맞출 수는 있지만 이것 또한 적절해 보이지 않음. 이는 높은 분산의 클래스이고 데이터의 과대적합이라고 함. 그럼 이 사이에 중간 단계의복잡함을 가지는 분류기가 있을 것인데, 두 번째 그래프 형태의 곡선이 데이터에 훨씬 더 적합해 보임. 이를 just right(딱 맞는 형태)라고 부름. 따라서 특성 x1과 x2만을 갖는 2차원의 예제에서는 데이터를 나타내고 편향과 분산을 시각화할 수 있음. 높은 차원의 문제에서는 데

이터를 나타내거나 결정 경계를 시각화할 수 없는데, 대신 편향과 분산을 이해하기 위해 살펴볼 몇 가지 기법들이 있음.

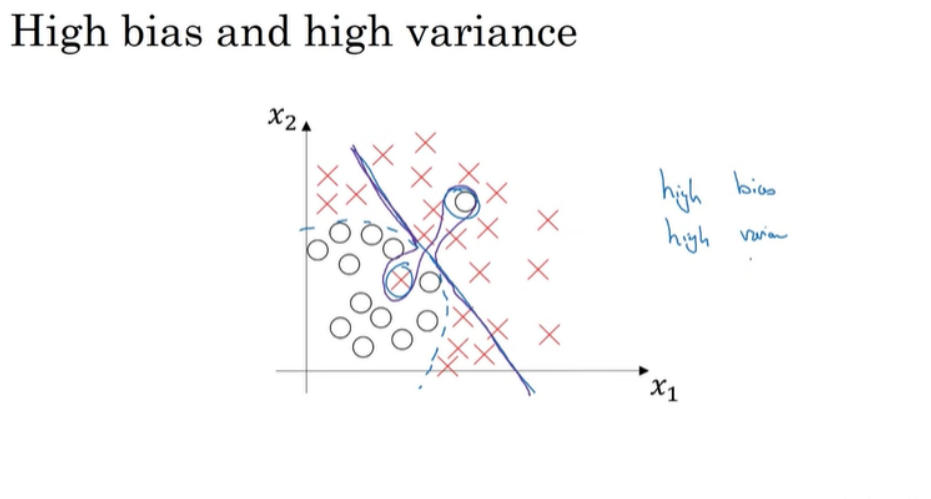


고양이 사진 분류 예제로 계속하면 여기 양성 샘플과 음성 샘플이 있는데, 편향과 분산을 이해하기 위한 중요한 두 가지 숫자는 훈련 세트 오차와 개발 세트 오차가 있음. 논증을 위해 고양이 사진을 인식하는 것은 사람들이 완벽히 할 수 있는 것이라고 가정했을 때, 훈련 세트 오차가 1%라고 하고 개발 세트 오차는 11%라고 함. 따라서 이 같은 예제에서 훈련 세트에서는 매우 잘 분류되었으나 상대적으로 개발 세트에서는 잘 분류되지 못한 것. 즉 훈련 세트에 과대적합 되어 개발 세트가 있는 교차 검증 세트에서 일반화되지 못한 경우임. 따라서 이런 경우의 예제는 높은 분산을 갖는다고 말함. 이렇게 훈련 세트 오차와 개발 세트 오차를 살펴봄으로써 알고리즘이 높은 분산을 갖는다는 것을 진단할 수 있게 됨.

이제 훈련 세트와 개발 세트 오차를 측정했는데 훈련 세트 오차가 15%, 개발 세트 오차가 16%라는 결과를 얻었다고 하면 이 경우 (인간은 대략 0%의 오차를 낸다고 가정) 이 알고리즘은 훈련 세트에서 조차 잘 작동하지 않으며 훈련 데이터에 대해서도 잘 맞지 않는다면 데이터에 과소적합한 것. 즉 이 알고리즘은 높은 편향을 가짐. 반면 이는 합리적인 수준의 개발 세트에서 일반화되고 있는데, 개발 세트의 성능이 훈련 세트보다 1%밖에 나쁘지 않기 때문. 이전 슬라이드의 맨 왼쪽 그림과 비슷한 경우임.

또 다른 예제로 훈련 세트 오차가 15%, 개발 세트 오차가 30%인 경우가 있는데, 이 알고리즘은 훈련 세트에 잘 맞지 않으므로 높은 편향을 지녔다고 할 수 있으며 또한 높은 분산을 가짐. 두 값 모두 최악인 경우. 마지막으로 훈련 세트 오차는 0.5%, 개발 세트 오차는 1%인 경우인데, 사용자는 오직 1%의 오차만 있는 고양이 분류기에 만족할 것. 이 분류기의 편향과 분산이 낮기 때문. 이 분석은 인간 수준의 성능이 거의 0%라는 가정에 근거하며, 더 일반적으로는 최적의 오차(베이지안 오차)가 거의 0%라는 가정임. 최적 오차가 더 높은 경우(예를 들어 15%), 훈련 세트 오차가 15%인 것은 아주 합당한 경우로 높은 편향이라 부르지 않고 낮은 분산이라고 할 수 있음. 어떤 분류기도 잘 작동하지 않을 경우 편향과 분산을 분석해야 하는데, 예를 들어 이

미지가 아주 흐릿해서 인간 혹은 그 어떤 시스템도 잘 분류하지 못할 경우 베이지안 오차는 훨씬 커질 것이고 이 분석에 대한 세부 방식은 달라질 것. 그러나 이 세부사항과는 별개로 중요한 것은 훈련 세트 오차를 확인함으로써 최소한 훈련 데이터에서 얼마나 알고리즘이 적합한지에 감을 잡을 수 있다는 것으로, 편향 문제가 있는지 알 수 있음. 훈련 세트에서 개발 세트로 갈 때 오차가 얼마나 커지냐에 따라서 분산 문제가 얼마나 나쁜지지에 대한 감을 잡을 수 있음. 훈련 세트에서 개발 세트로 일반화를 잘 하느냐에 따라 분산에 대한 감이 달라짐. 이 모든 것은 베이즈 오차가 매우 작고, 훈련 세트와 개발 세트가 같은 확률 분포에서 왔다는 가정하에 이루어짐. 이 가정들이 지켜지지 않는 다면 더 복잡한 분석 방법을 사용해야 함.



이전 슬라이드에서 높은 편향 또 높은 분산일 때 그래프 모양을 확인했었는데, 이번에는 높은 편향 & 높은 분산의 경우를 살펴볼 것(최악의 경우). 선형의 분류기는 데이터에 과소적합하기 때문에 높은 편향을 가짐. 따라서 이 분류기(보라색_)는거의 선형이고 데이터에 과소적합할 것. 그러나 만약 분류기가 이상하게 동작해서 일부의 데이터에 과대적합한다면 보라색 분류기는 높은 편향과 높은 분산을 갖게됨. 선형의 분류기는 이런 2차 곡선에 맞지 않으므로 높은 편향을 가짐. 그러나 중간에 너무 많은 굴곡을 가져서 특정 샘플들이 과대적합 됨. 따라서 해당 분류기는 거의 선형이지만 곡선이나 이차함수가 필요하기 때문에 높은 편향을 가짐. 또한 중간에 잘못 라벨링된 샘플을 맞추기 위해 너무 많은 굴곡을 갖기 때문에 높은 분산을 갖게 되는됨. 이 예제는 이차원에서 고안된 예제임. 매우 높은 차원의 입력에서는 어떤 영역은 높은 편향을 가지고 어떤 영역은 높은 분산을 갖게될 것. 따라서 높은 차원의 입력에서도 이러한 모습이 나타날 수 있음.

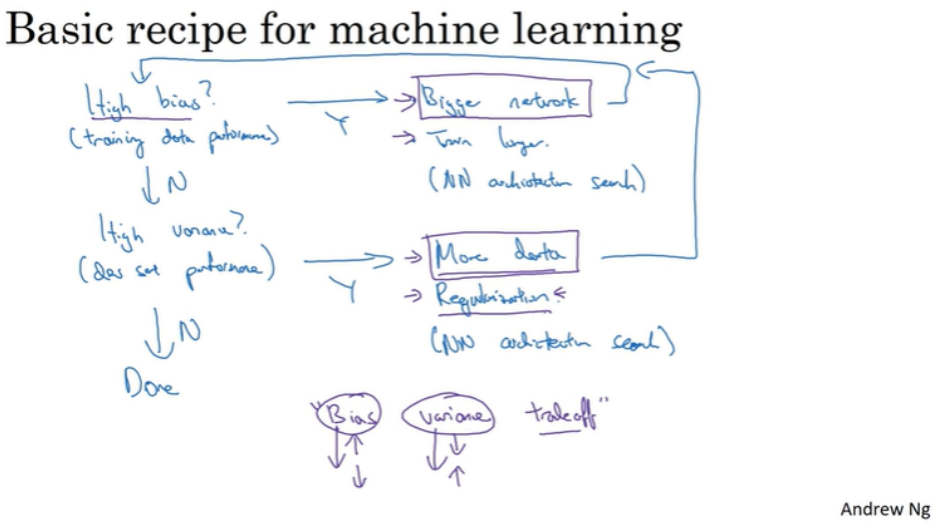
🔎 참고자료

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

	높은 분산 (과대적합)	높은 편향 (과소적합)	높은 편향 & 높은 분산	낮은 편향 & 낮은 분산
훈련 세트	1 %	15 %	15 %	0.5 %
개발 세트	11 %	11 %	30 %	1 %

📌머신러닝을 위한 기본 레시피(C2W1L03)

핵심어: 편향(Bias), 분산(Variance)



Andrew Ng

최초의 모델을 훈련하고 난 뒤 처음 질문하는 것은 알고리즘이 높은 편향을 가지는지이며, 높은 편향을 평가하기 위해서는 훈련 세트 혹은 훈련 데이터의 성능을 봐야 함. 만약 높은 편향을 가져서 훈련 세트에도 잘 맞지 않는다면, 더 많은 은닉 층 혹은 은닉 유닛을 갖는 네트워크를 선택하는 것임. 또는 더 오랜 시간 훈련시키거나 다른 발전된 최적화 알고리즘을 사용할 수도 있음. 다른 방법으로는 (작동할 수도 아닐수도 있지만) 나중에 살펴볼 내용으로 다양한 신경망 아키텍처가 있음. 즉 이 문제에 더 잘 맞는 신경망 아키텍처를 찾을 수도 있는 것. 더 큰 네트워크를 갖는 것은 대부분 도움이 되고, 더 오래 훈련시키는 것은 도움이 안 될 수도 있지만 해는 되지 않음. 따라서 학습 알고리즘은 훈련시킬 때 최소한 편향 문제를 해결할 때까지 이 방법들을 시도할 것. 보통 충분히 큰 네트워크라면 보통은 훈련 데이터에 잘 맞출 수 있음. 어떤 경우에는 가능하지만 이미지가 매우 흐릿한 경우는 불가능할 수 있으나, 최소한 인간

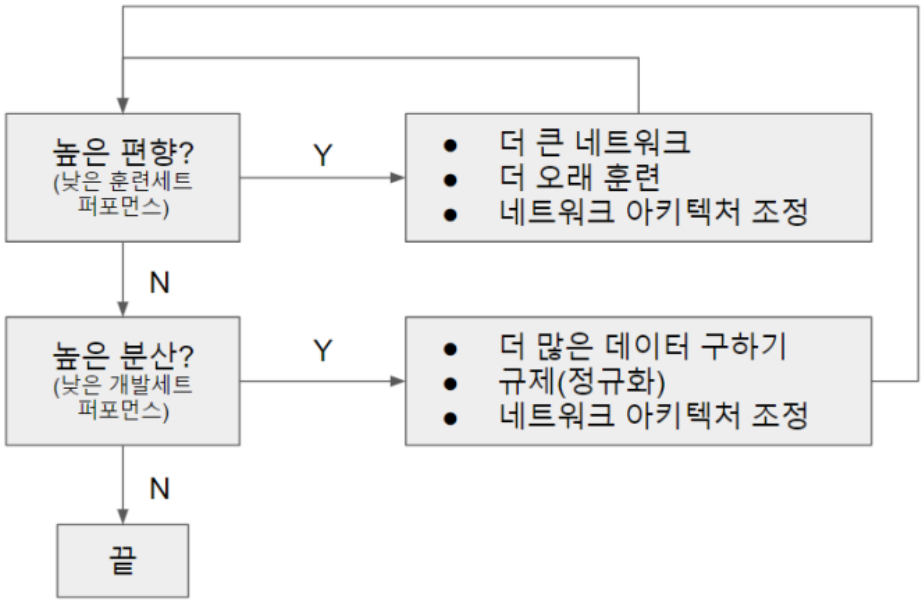
이 구별할 수 있다면, 다시 말해 베이즈 오차가 매우 작다면 그것보다 더 크게 훈련하는 경우 최소한 훈련 세트에 대해서는 잘 맞을 것(과대적합이 되더라도). \

편향을 수용 가능한 크기로 줄이게 되면 그 다음에 물어볼 것은 분산 문제가 있는지인데, 이를 평가하기 위해 개발 세트 성능을 보게됨(꽤 좋은 훈련 세트 성능에서 꽤 좋은 개발 세트 성능을 일반화할 수 있는지). 높은 분산 문제가 있을 때 이를 해결하는 가장 좋은 방법은 데이터를 더 얻는 것인데, 얻을 수 있다면 가장 좋은 방법임. 그러나 가끔 데이터를 더 못 얻는 경우도 있는데, 이때는 과대적합을 줄이기 위해 정규화를 시도하거나 다른 신경망 아키텍처를 찾는 것을 시도해볼 수도 있음. 더 적합한 신경망 아키텍처를 찾을 수 있다면 가끔은 그것이 앞의 편향 문제처럼 분산을 줄일 수 있음. 그러나 이 방법을 완전히 체계화하기는 어려움. 이 방법들은 낮은 편향과 분산을 찾을 때까지 계속 시도하고 반복하게 됨.

몇 가지 중요한 것은, 첫 번째로 높은 편향이냐 분산이냐에 따라 시도해 볼 수 있는 방법이 아주 달라질 수 있다는 것. 그래서 주로 훈련과 개발 세트를 편향이나 분산 문제가 있는지 진단하는데 사용하며 그 결과 시도해볼 수 있는 방법을 적절하게 선택함. 예를 들어 노은 편향 문제가 있다면, 더 많은 훈련 데이터를 얻는 것은 크게 도움이 되지 않으며 가장 효율적인 방법이 아님. 따라서 편향과 분산, 혹은 둘 다의 문제가 얼마나 있는지를 명확하게 하는 것은 가장 유용한 시도를 선택하는 데 집중할 수 있도록 함. 두 번째로 초기 머신러닝의 시대에는 편향-분산 트레이드오프에 대한 많은 논의가 있었는데, 그 이유는 시도할 수 있는 많은 것들이 편향 증가시키고 분산을 감소시키거나 편향을 감소시키고 분산을 증가시키기 때문. 딥러닝 이전 시대로 돌아가면 툴이 그렇게 많지는 않은데, 서로를 나쁘게 하지 않고 편향만을 감소시키거나 분산만 감소시키는 툴이 많이 없었음.

그러나 현대의 딥러닝 빅데이터 시대에는 더 큰 네트워크를 훈련시키고 더 많은 데이터를 얻는 것이 둘 다의 경우에 (항상 적용되는 것은 아니지만), 더 큰 네트워크를 갖는 것이 대부분 분산을 해치지 않고 편향만을 감소시킴(정규화를 올바르게 했을 경우). 그리고 데이터를 더 얻는 것도 대부분 편향을 해치지 않고 분산을 갑소시킴. 따라서 더 큰 네트워크를 훈련시키거나 더 많은 데이터를 얻는 이 두 단계는 편향만을 감소시키거나 분산만을 감소시키는 툴이 됨(서로 영향을 미치지 않고). 이것이 지도학습에 딥러닝이 매우 유용한 큰 이유 중 하나라고 볼 수 있는데, 편향과 분산의 균형을 신경써야 하는 트레이드오프가 훨씬 적기 때문임. 그러나 가끔은 편향이나 분산을 줄이는데 어쩔 수 없이 다른 것을 증가시키는 선택을 하는 경우도 있음. 사실 최종적으로 잘 정규화된 네트워크를 갖는 방법도 있는데, 이는 추후에 다룰 예정. 결론은 더 큰 네트워크를 훈련시키는 것이 대부분 절대 해가 되지 않으며, 너무 큰 신경망을 훈련시키는 주된 비용은 계산 시간이라는 것(정규화도 마찬가지).

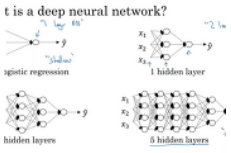
🔎 **참고자료** (위 슬라이드를 도식화)



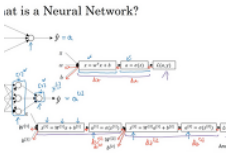
공감

'**DL 스터디&프로젝트**' 카테고리의 다른 글

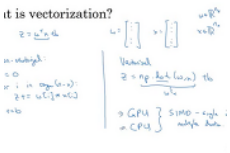
관련글



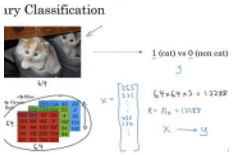
[Euron 중급 세션...



[Euron 중급 세션...



[Euron 중급 세션...



[Euron 중급 세션...

댓글 0

공부하자_

내용을 입력하세요.

등록