

Deep Learning Ch1-2

CH1

딥러닝이란 무엇인가?

Deep-Learning

현재 AI 역사에서 찾을 수 없을 만큼 대중에게 많은 관심과 업계의 투자 받음

머신 러닝의 첫 번째 성공 X

오늘날 산업계에서 사용하는 대부분의 머신 러닝 알고리즘!=딥러닝 알고리즘

때론 딥러닝 적용하기에 데이터 충분 X/다른 알고리즘 문제 해결에 더 유용

=> 딥러닝이 모든 작업에 맞는 만능 도구 X, 따라서 전통적 머신 러닝 방법에 대해

간단하게 소개, 지금까지 역사적 배경 설명

1.2.1

확률적 모델링

(probabilistic modeling)

통계학 이론을 데이터 분석에 응용한 것
초창기 머신 러닝 형태 中 1이며, 요즘도 널리 사용

나이브 베이즈 알고리즘 : 가장 잘 알려진 알고리즘,
입력 데이터 특성이 모두 독립적이라 가정+베이즈
정리를 적용하는 머신 러닝 분류 알고리즘

=> 이런 형태의 데이터 분석은 컴퓨터보다 앞서있어
첫 번째 컴퓨터가 등장하기 전엔 수작업으로 적용

* 로지스틱 회귀

현대 머신 러닝의 "hello world"로 여겨짐, 회귀 알고리즘이 아니라 *분류 알고리즘
나이브 베이즈와 매우 비슷하게 컴퓨터보다 훨씬 오래 전부터 존재
간단+다목적 활용 가능한 특성 덕분에 여전히 유용하게 사용

=> 데이터 과학자가 분류 작업에 대한 감을 빠르게 얻기 위해 데이터셋에 적용할
첫 번째 알고리즘으로 선택하는 경우 多

**회귀는 연속적인 숫자(실수)를 예측하는 것이고, 분류는 여러 클래스 중 하나를 예측하는 것*

1.2.2

초창기 신경망

1950년대 : 대규모 신경망 훈련시킬 수 있는 효과적인 방법을 오랜 기간 동안 찾지 못했기 때문에 신경망 핵심 아이디어가 작게나마 연구되는데 수십 년 걸림

1980년대 중반 : 여러 사람이 각각 **역전파 알고리즘 재발견**, 신경망에 이를 적용하며 상황 변화

=> 이 알고리즘은 경사 하강법 최적화를 사용하여 연쇄적으로 변수가 연결된 연산 훈련 방법

* 성공적인 첫 번째 신경망 애플리케이션

1989년 벨 연구소에서 나옴, 얀 르쿤이 초창기 합성곱 신경망 + 역전파 연결하여 손글씨 숫자 이미지 분류하는 문제에 적용

=> **LeNet 신경망**이라 불리며, 우편 봉투의 우편 번호 코드를 자동으로 읽기 위해 1990년대 미국 우편 서비스에 사용

1.2.3

A1. 커널 방법 (Kernel Method)

90년대에 신경망 이후, 커널 방법이 인기

커널 방법 : 분류 알고리즘의 한 종류, *서포트 벡터 머신(SVM)이 제일 유명

오래된 선형 공식 : 바프닉, 알렉세이가 개발,
63년 공개

현대적 SVM 공식 : 90년대 초 벨 연구소에서 개발,
95년 공개

*SVM은 분류뿐만 아니라 회귀 문제에도 사용 가능

* SVM

분류 문제를 해결하기 위해 2개의 다른 범주에 속한 데이터 포인트 그룹 사이에
좋은 결정 경계를 찾음

결정 경계 : 훈련 데이터를 2개의 범주에 대응하는 영역으로 나누는 직선이나 표면

=> 새로운 데이터 포인트를 분류하려면 결정 경계 어느 쪽에 속하는지 확인하면 됨

* SVM이 결정 경계 찾는 과정

① 결정 경계가 하나의 초평면으로 표현될 수 있는 새로운 고차원 표현으로 데이터를 매핑

② 마진 최대화 : 초평면과 각 클래스의 가장 가까운 데이터 포인트 사이의 거리가 최대가 되는
최선의 결정 경계(하나의 분할 초평면)을 찾는다

=> 이를 통해 결정 경계가 훈련 데이터셋 이외의 새로운 샘플에 잘 일반화되도록 도움

1.2.3

A1. 커널 방법 (Kernel Method)

90년대에 신경망 이후, 커널 방법이 인기

커널 방법 : 분류 알고리즘의 한 종류, *서포트 벡터 머신(SVM)이 제일 유명

오래된 선형 공식 : 바프닉, 알렉세이가 개발,
63년 공개

현대적 SVM 공식 : 90년대 초 벨 연구소에서 개발,
95년 공개

**SVM은 분류뿐만 아니라 회귀 문제에도 사용 가능*

* SVM

분류 문제를 해결하기 위해 2개의 다른 범주에 속한 데이터 포인트 그룹 사이에
좋은 *결정 경계를 찾음, 수학적으로 깊게 분석하기 용이해서 이론 이해/설명 쉬운 특징
=>머신 러닝 분야에서 오랫동안 큰 인기

**결정 경계 : 훈련 데이터를 2개의 범주에 대응하는 영역으로 나누는 직선이나 표면*
=> 새로운 데이터 포인트를 분류하려면 결정 경계 어느 쪽에 속하는지 확인하면 됨

* SVM이 결정 경계 찾는 과정

- ① 결정 경계가 하나의 초평면으로 표현될 수 있는 새로운 고차원 표현으로 데이터를 매핑
- ② 마진 최대화 : 초평면과 각 클래스의 가장 가까운 데이터 포인트 사이의 거리가 최대가 되는 최선의 결정 경계(하나의 분할 초평면)을 찾는다

=> 이를 통해 결정 경계가 **훈련 데이터셋 이외의 새로운 샘플에 잘 일반화**되도록 도움

* SVM의 단점

- ① 대용량 데이터셋에 확장되기 어렵
- ② 이미지 분류 같은 지각 관련 문제에서 좋은 성능 X
- ③ 얇은 학습 방법, 지각 관련 문제 적용하려면 수동으로 유용 표현 추출(=특성 공학)해야 하는데, 이는 매우 어렵고 불안정

1.2.3

A2. 커널 방법 (Kernel Method)

90년대에 신경망 이후, 커널 방법이 인기

커널 방법 : 분류 알고리즘의 한 종류, *서포트 벡터 머신(SVM)이 제일 유명

오래된 선형 공식 : 바프닉, 알렉세이가 개발,
63년 공개

현대적 SVM 공식 : 90년대 초 벨 연구소에서 개발,
95년 공개

*SVM은 분류뿐만 아니라 회귀 문제에도 사용 가능

* 커널 기법

새롭게 표현된 공간에서 좋은 결정 초평면 찾기 위해 새로운 공간 대응하는
데이터 포인트의 좌표 실제로 구할 필요 X

새로운 공간에서 두 데이터 포인트 사이의 거리를 계산할 수 있으면 됨

=> 커널 함수를 사용하면 이를 효율적으로 계산 가능

커널 함수 : 원본 공간에 있는 두 데이터 포인트를 명시적으로 새로운 표현으로 변환하지 않고
타깃 표현 공간에 위치했을 때의 거리를 매핑해주는 계산 가능한 연산

=> 커널 함수는 일반적으로 데이터로부터 학습되지 않고 직접 만들어야 함,
SVM에서 학습되는 것은 분할 초평면뿐

1.2.4

A. 결정 트리 (decision tree)

플로차트 같은 구조를 가지며 입력 데이터 포인트를
분류 or 주어진 입력에 대해 **출력 값 예측**

결정 트리는 **시각화하고 이해하기 쉬움**

데이터에서 학습되는 결정 트리는 2000년대부터
연구자들에게 크게 관심을 받기 시작했고,
2010년까지는 커널 방법보다 선호

B. 랜덤 포레스트 (Random Forest)

결정 트리 학습에 기초하여 안정적, 실전에서 유용

서로 다른 결정 트리 많이 만들고 그 출력을 **앙상블**
하는 방법 사용

=> 다양한 문제에 적용 가능, 얇은 학습 해당하는
어떤 작업에서도 거의 항상 두 번째로 좋은 알고리즘

C. 그래디언트 부스팅 머신 (gradient boosting machine)

랜덤 포레스트와 매우 유사, 약한 예측 모델인 결정
트리 앙상블하는 것을 기반으로 하는 머신 러닝 기법

그래디언트 부스팅 : 이전 모델에서 놓친 데이터
포인트를 보완하는 새로운 모델 반복 훈련함으로써
머신 러닝 모델 향상하는 기법

결정 트리에 그래디언트 부스팅 기법 적용
=> *랜덤 포레스트와 비슷하지만 대부분 이 성능을
뛰어넘는 모델을 만들*

1.2.5

다시 신경망으로

2010년경 신경망 대부분 과학 커뮤니티에서 관심 받지 못했지만, 여전히 신경망 대해 연구하고 있던 일부 사람들이 중요한 성과 내기 시작

2011년 IDSIA의 댄 크리슨이 GPU로 훈련된 심층 신경망으로 학술 이미지 분류 대회에서 우승한 것이 시작
=> 현대적인 딥러닝의 첫 번째 성공

2012년 ImageNet에 힌튼 팀이 1,400만 개의 이미지를 훈련시킨 후, 고해상도 컬러 이미지를 1,000개 범주로 분류하는 어려운 문제를
전년도 74.3% -> 83.6% 달성

=> 이때부터 심층 합성곱 신경망이 계속 우승

2015년에는 96.4% 달성하며 ImageNet 분류 문제 완전 해결로 간주

=> 심층 합성곱 신경망이 모든 컴퓨터 비전 작업의 주력 알고리즘이 됨
해당 알고리즘은 조금 더 **일반적+지각에 관한 모든 문제에 적용 가능**

1.2.6

딥러닝의 특징

빠르게 확산된 이유 : 많은 문제에서 더 좋은 성능
=> 머신 러닝에서 가장 중요한 단계인 **특성 공학**을 **완전히 자동화**하여 문제를 더 해결하기 쉽게 만듦

특성 공학 : 데이터 좋은 표현을 수동으로 만드는 것,
=> 처리하기 용이하게 사람이 초기 입력 데이터를 여러 방식으로 변환해야 하기 때문

딥러닝 : 이를 완전히 자동화 + 특성 직접 찾는 대신 한 번에 모든 특성 학습

=> 머신 러닝 작업 흐름 단순화

*** 얇은 학습 방법도 딥러닝 효과 모사하기 위해 반복 적용 가능하지 않을까?**

A. 실제로 얇은 학습 방법 연속 적용하면 각 층 효과 빠르게 줄어듦,
3개의 층을 가진 모델에서 최적의 첫 번째 표현 층은 1 or 2개의 층을 가진 모델에서 최적의 첫 번째 층과 달라야 함

*** 딥러닝 변환 능력**

모델이 모든 표현 층을 순차적이 아니라 동시에 공동 학습하게 함

=> 이 덕분에 모델이 내부 특성 하나에 맞추어질 때마다 이에 의존하는 다른 모든 특성이 사람이 개입하지 않아도 자동으로 변화에 적응하게 되며, 모든 학습은 하나의 피드백 신호에 의해 시작
=> 모델의 모든 변화는 최종 목표를 따라가게 됨

이는 모델을 많은 층으로 나누어 복잡하고 추상화 된 표현을 학습 시킬 수 있어, 얇은 학습 모델을 탐욕적으로 쌓은 것보다 강력
이때, 각 층은 이전 층에 의존하지 않는 단순한 변환 수행

*** 딥러닝이 데이터로부터 학습하는 방법에는 두 가지 중요한 특징**

① 층을 거치면서 점진적으로 더 복잡한 표현 만들어진다

② 이런 점진적인 중간 표현이 공동으로 학습된다 각 층은 상/하위 층의 표현이 변함에 따라 함께 바뀜

=> 이 2개의 특징이 이전의 머신 러닝 접근 방법보다 딥러닝이 훨씬 성공하게 된 이유

1.2.7

머신 러닝의 최근 동향

캐글의 머신 러닝 경연을 살펴본다면, 머신 러닝의 최근 동향 알기 좋음은 문제에서 더 좋은 성능

그래디언트 부스팅 : 구조적인 데이터인 경우 사용, XGBoost 라이브러리 사용 (파이썬, R을 지원)

딥러닝 : 이미지 분류 같은 지각 관한 문제에 사용, Keras 라이브러리 사용 (파이썬 지원, 쉽고 유용)

머신 러닝을 성공적으로 적용하기 위해 알아 할 두 가지 기술

- ① 얇은 학습 문제를 위한 그래디언트 부스팅 머신
- ② 지각에 관한 문제를 위한 딥러닝

=> 기술적으로 보면 캐글 경연 대회에서 주로 사용되는 **XGBoost**와 **Keras**를 배워야 함

Summary

딥러닝+전통 머신 러닝 방법에 대한 간략 설명

- ① 로지스틱 회귀 : 확률적 모델링 중 하나이며, 나이브 베이즈와 함께 가장 잘 알려져있음
 - ② 초창기 신경망 : 역전파 알고리즘 재발견, LeNet 신경망이라 불리며 과거 활발하게 사용
 - ③ 커널 방법 : SVM이 가장 유명하며, 새로운 공간에서 두 데이터 포인트 사이 거리 계산할 때, 유용하게 사용
 - ④ 결정 트리 : 주어진 입력 값에 대한 출력 예측, 이를 이용한 랜덤 포레스트/그래디언트 부스팅 머신
 - ⑤ 다시 신경망 : 재조명 받기 시작하며, 심층 합성곱 신경망이 주력 알고리즘이 됨
- => 현재, 그래디언트 부스팅과 딥러닝이 가장 많이 사용됨

THANK
YOU