딥러닝과 데이터

데이터는 기하급수적으로 늘어났다. 단위 연산당 비용은 엄청나게 줄어들었다. 그 결과 인공 신경망 기반의 기계 학습 분야가 각광받고 있다. 과거 인공 신경망은 다른 기계 학습 방법론들에 비해 여러 단점[*1]을 가지고 있었다. 그러나 21세기 들어 많은 문제들이 해결되었다. 다수의 은닉층(hidden layer) 기반 심층 인공 신경망[*2]은 1990년대에는 시도조차 할 수 없었다. 심층 신경망은 사전 지식 없이 데이터로부터 통찰을 얻어 내거나 더 나아가 인간이 통찰을 얻기 어려운 데이터를 대상으로도 일정 정도의 처리를 해내는 능력을 보였다. 이는 인간이 직관적으로 접근하기 어려운 거대 데이터 기반의 분석 및 특징 추출을 종단간 모형[*3]으로 해결할 수 있다는 것을 의미한다. 이러한 이유로 심층 신경망 분야에 대한 주목도가 계속 높아지고 있다.

그러나 응용 환경에서 종단간 모형 기반의 딥러닝 모형을 도입하는 것은 어렵다. 가장 큰 제약은 시간과 비용이다. 종단간 심층 신경망의 경우 원하는 결과를 얻기 위해서 엄청난 양의 데이터 및 연산 자원이 필요하다. 충분히 깊은 심층 신경망의 경우 입력층에 가까운 계층들이 데이터 전처리를 담당하도록 훈련되는 경향이 있다. 그러나 데이터 전처리를 위해 은닉 계층을 늘릴수록 신경망의 복잡도가 크게 증가한다[*4*5]. 또한 은닉층의 수가 늘어날수록 훈련 과정에서 수렴 상태에 도달하기 위해 더 많은 데이터가 필요하다. 이러한 문제는 모형 개발 과정에서의 디버깅(debugging)의 어려움, 훈련 과정의 막대한 시간 및 자원 소모와 함께 그 결과로 얻은 비대화된 모형을 사용할 때 발생하는 추론 비용의 증가로 이어진다.

글 | 신정규 jshin@lablup.com

노는게 제일 좋아 친구들 모여라. 언제나 즐거워 삽질쟁이 신정규.코드덮힌 삽질 마을 쪼렙아빠 나가신다. 언제나 즐거워 오늘은 또 무슨 일이 생길까. 머신러닝 훈련 및 추론용 분산처리 프레임워크를 개발하는 래블업 주식회사 대표. 두뇌 및 사회 시스템의 의견형성동역학(opinion formation dynamics)을 연구하는 쪼렙 물리학자. 오픈소스 옹호자. 텍스트큐브 개발자. TNF/니들웍스. 꿈꾸는 사람.

빅데이터 처리에 중요하게 간주되었던 데이터 전처리 및 결과의 후처리 과정은 인공 신경망 기반의 기계 학습 모형 설계 과정에서도 여전히 매우 중요하다. 기계 학습 모형이 '정해진 시간 안에' '제대로 된 결과'를 내놓을 수 있게 돕기 때문이다. 데이터 전처리를 통해 잘 정의되고 정제된 데이터와 특징(feature)을 사용하면 전체 신경망의 크기 및 복잡도를 줄일 수 있다. 또한 결과의 후처리는 멀티 모달 모형(multi modal model)[*6] 설계 시 모델 간의 연결에 중요한 역할을 담당한다.

그런데 인공 신경망 훈련을 위한 데이터 전처리 과정에서는 일반적인 데이터 분석을 위한 전처리 과정에 더하여 여러 가지를 고려해야 한다. 이 글에서는 인공 신경망 훈련을 위한 데이터 전처리 과정에서 고려해야 할 요소들을 실제 경험한 사례들과 함께 짚어 보겠다.

동일한 현상에서 얻은 동일하지 않은 데이터: 정규화의 함정

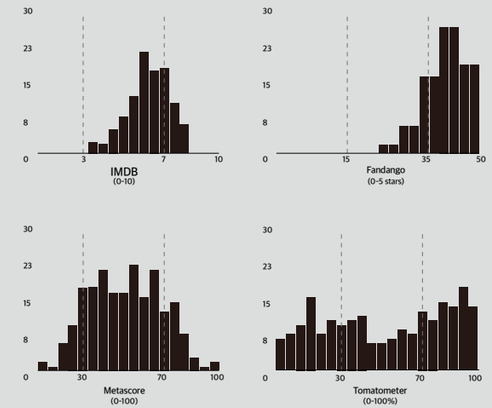
미디어 추천 시스템을 만드는 경우를 가정하자. 개인화 추천 시스템으로, 어떤 사용자가 어떤 콘텐츠를 얼마나 좋아할 것인지를 예측하는 모형을 만드는 것이 목표이다. 모형의 훈련 데이터로 가장 쉽게 사용할 수 있는 것은 랭킹 데이터이다. 수많은 사용자들이 영화 및 드라마에 점수를 매겨 놓은 랭킹 데이터를 가정해 보자. 네이버 영화 평점은 10점 만점 시스템, 왓챠의 시스템은 5점 만점 별표 시스템이다. (중간에 별표 반 개를 가능하게 하여 10점 시스템으로 바뀌었지만, 이러한 경우는 뒤에서 따로 다룰 것이므로 여기에서는 논외로 한다) 넷플릭스의 경우 이진 평점(좋아요/아니요)이다.

이 세 가지 종류의 데이터를 다 갖고 있을 경우 전처리를 거쳐 동일한 데이터 세트를 만든 후 훈련하는 데 사용할 수 있을까? 답부터 이야기하자면 불가능하다.

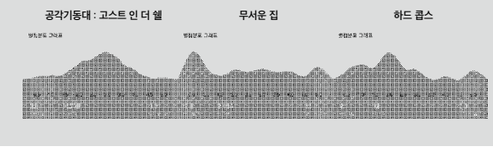
서로 다른 스케일의 데이터를 정규화하여 하나의 데이터 세트로 만들어 보자. 어느 데이터 세트를 기준으로 스케일을 맞출 것인가? 아티팩트(artifact)를 추가하지 않기 위해서는 더 낮은 해상도의 데이터 세트로 스케일하는 것이 일반적이다. 그렇다면 위에서 예로 든 데이터 세트의 경우 좋아요/아니요의 이진 데이터로 스케일해야 한다. 이 경우 10점 만점의 데이터는 몇 점을 기준으로 좋아요/아니요로 변환해야 할까? 만약 데이터가 바이모달 분포(bimodal distribution)[*7]를 따르고, 최고점이 두 개라면 나누기 쉬울 것이다. 하지만 유니모달(unimodal)이거나, 또는 멀티모달인데 최고점이 여러 개라면 어떤 기준으로 데이터를 분류해야 할까?

일단 임의의 기준으로 평점 데이터를 이진 데이터로 변환하고 적절한 기계 학습 모형을 만들어 훈련시켜 보자. 기계 학습 훈련 과정에 데이터 정규화 과정이 끼치는 영향은 엄청나다. 상위 50%는 좋아요, 하위 50%는 싫어요로 변환한 데이터로 훈련한 경우와, 상위 52%는 좋아요, 하위 48%는 싫어요로 변환한 데이터로 훈련한 경우의 기계 학습 모형은 동일한 입력에 대해 상당히 다른 추론 결과를 내놓는다[*8].

[그림 1] 평점 데이터들의 분포[*9]



[그림 2] 평점 데이터는 기본적으로 이산 분포 사이를 트렌드로 채운 값이다. 유니모달, 바이모달, 멀티모달의 예[*10]



수학적으로는 문제가 될 수 있지만 편의상 이진 기준이 아닌 선형 스케일로 데이터를 스케일하여 포맷을 맞출 경우를 생각해 보자. 10점 기준으로 맞출 경우 5점 기준과 이진 기준의 평가는 전체 데이터에 완전히 편향된 경향을 추가하게 된다. 5점 기준으로 맞출 경우 10점 데이터의 앨리어싱 기준이 문제가 된다. 더 본질적인 문제가 있다. 랭킹의 경우 인간이 능동적으로 매기는 라벨이다. 10점 만점의 4점과, 5점 만점의 2점은 심리적으로 다른 반응을 불러일으킨다. 따라서 실질적으로는 다른 데이터라 단순 스케일로 맞출 수 없을 것이다[*11*12].

이런 문제를 해결하는 가장 간단한 방법은 애초에 논란이 생기지 않을 데이터를 생성하는 것이다. 몇 가지 실험 후 넷플릭스(Netflix)는 2017년 봄부터 이진 평점만을 사용하고 있다[*13]. 프로필 데이터의 해상도 감소를 감수하고서라도 가공 및 훈련을 원활하게 하기 위한 선택이다. 오래전 구글의 동영상 서비스인 유튜브(Youtube)는 평점 유효성 문제(대부분의 사람이 5점 아니면 1점만 주는)로 마찬가지의 선택을 하였다[*14].

동일한 현상, 다른 데이터

IT 시스템에서 생성된 데이터는 균일하다는 일반적인 믿음이 있다. 이 믿음은 무거운 물체가 빨리 떨어질 것이라는 직관과 비슷하다. IT 인프라스트럭처는 업그레이드가 가장 빠른 분야 중 하나다. 시스템에서 생성되는 데이터는 동일한 현상을 다루고 있어도 다른 데이터를 만들어 낸다. 가장 일반적으로 접할 수 있는 것은 로그 시스템이나, 로그 정책이 바뀌는 경우들이다. 시스템 업그레이드 시 다루는 메트릭의 종류 및 속성이 바뀌는 경우도 빈번하다.

채팅을 하는 기계 학습 모형(chatbot, 챗봇)을 만든다고 가정하자. 상업적으로 챗봇을 만들려고 시도하는 기업들은 대부분 고객 응대 분야에서 다년간 축적한 데이터를 소유하고 있다. 이 데이터로 챗봇 모형을 만들 수 있을까? 보통은 불가능하다. 일반적인 상담 로그 데이터들은 중간에 몇 번의 형식 변경을 거친 데이터들이다. 또한 다양한 상담 환경에서 작성된 데이터들이기도 하다. 엄청난 전처리 과정이 필요하다.

기록 방식의 변경뿐 아니라, 데이터를 만드는 인프라의 영향 또한 고려해야 할 요소이다. 생명과학 및 헬스케어 스타트업에서 특이 유전자 분석 과정을 처리하는 기계 학습 모형을 만드는 작업 흐름을 가정해 보자[*15]. 고객 표본에서 추출한 RNA를 대량으로 뻥튀기하고, 유전자 칩[*16]을 이용해 유전 패턴의 이상발현 여부를 찾는다[*17]. 특정 유전 패턴이 정상보다 더 많이 발현되거나 덜 발현된 경우, 유전자 칩 이미지의 픽셀 강도 차이로 나타난다. 이 이미지들을 모아 CNN기반의 모형을 훈련한다. 훈련이 끝난 모형을 이용하여 특정 질병들의 발병 여부를 한 번에 찾아내는 분류자로 사용할 수 있을 것이다. 작동할까? 데이터가 올바르다면 어느 정도의 성과가 있을 것이다.

분석 기기로부터 데이터를 측정하여 모형 훈련을 위한 데이터를 만들어야 할 것이다. 유전자 칩 및 분석 기기를 만드는 회사로는 일루미나(Illumina) 및 에피메트릭스(Affymetrix) 등이 있다. 각 회사의 기기를 반반씩 구입하면 구입 예산의 반을 날리는 경험을 할 수 있다. 두 기기는 동일한 실험을 했을 때에도 서로 다른 이상발현 유전자를 지목한다[*18*19]. (주로 특허로 인한) 다른 기기 설계, 다른 데이터 획득 방법, 데이터 전처리 등 기기 전반에 걸친 차이가 누적되어 이러한 차이를 만든다. 두 시스템에서 만들어 낸 실험 결과를 섞어서 기계 학습 모형을 훈련하면 실제 데이터 대상으로 사용할 수 없는 모형이 만들어진다.

실험 기기들에서 원시 데이터[*20]를 추출해 데이터베이스를 만든 경우에도 모형은 학습되지 않을 것이다. 유전자 칩 정도[*21]의 데이터를 뽑아내는 기기들의 경우, 엄밀한 의미에서의 원시 데이터는 존재하지 않기 때문이다. 생명과학 실험 장비들은 대상의 특성상 노이즈가 엄청난 데이터를 측정한다[*22]. 이 데이터를 그대로 내보낼 경우에도 기기가 일반적인 통계 전처리를 수행한다.

위의 문제에 대한 가장 간단한 해결 방법은 동일한 현상에 대해 동일한 데이터를 얻을 수 있는 환경을 만드는 것이다. 챗봇 모형 개발의 경우 데이터 형식 통일 작업, (음성 또는 문자 등의) 상담 환경에 따른 분류 작업, 상담 카테고리에 따른 분류 작업 등을 거쳐 데이터 포맷을 맞춘다. 그 후 방언 제거, 은어 치환, 상담 요청자의 문장 길이에 따른 정렬[*23]을 거쳐 전처리 데이터를 완성하는 것이 일반적인 과정이다. 유전자 분석 모형의 경우 한 공급처에서 측정 기기를 구입해야 하고, 데이터 후처리 과정에서는 공급사가 제공한 도구 키트 대신 원시 데이터를 꺼내서 전처리 과정을 자체 구축하여 데이터를 다듬어야 할 것이다[*24].

젊은 '빅'데이터 : 시간축에 따른 데이터 밀도차의 문제

패션 데이터를 모아 트렌드에 따른 패션을 제안하는 기계 학습 모형을 설계해 보자[*25]. 우선 패션의 적합도를 알려 주는 모형을 만들어야 할 것이다. 패션 모형의 훈련을 위한 다양한 데이터를 획득했다고 하자[*26]. 이 모형은 충분한 데이터가 있다면 트렌드를 예측할 수 있을까? 그럴 수도 있고 그렇지 않을 수도 있다. 보통은 다양한 편향의 영향으로 모형이 제대로 동작하지 않을 것이다. 편향은 시간 의존적인 데이터 밀도 차이에서 비롯되기도 한다.

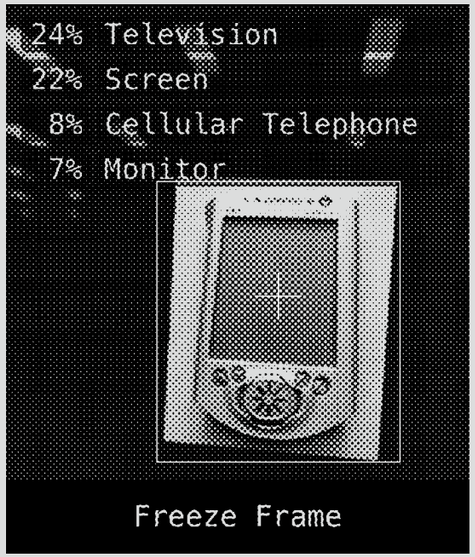
심층 신경망의 대두에는 심층 신경망을 훈련할 수 있는 충분한 (엄청난) 양의 데이터가 뒷받침되었다. 그런데 그 데이터들이 어디에서 왔을까? 사실 '어디'보다는 '언제'가 더 적합한 질문이다. 거의 모든 빅데이터는 최근에 생성되었다. 빅데이터는 '더 다양한' 데이터를 '생성'하고 '기록'하는 과정을 전산화하는 과정의 부산물이다. 그런데 빅데이터의 증가 추세는 지수적 증가에 가깝다. 비교적 오래되고 계량화된 주식 거래 데이터의 경우를 살펴보자. 뉴욕 증권 거래소의 1993년 거래 틱 데이터의 총 용량은 4.25기가이다. 1998년에는 20기가가 되었고, 2001년에는 90.9기가, 2004년에는 455기가가 되었다[*27]. 단지 1년의 차이로도 누적되는 데이터의 크기가 달라진다.

[그림 3] 연도별 NYSE 틱데이터 데이터 크기의 추세

이러한 데이터 밀도 차는 최종 모형의 추론 과정에서 시간에 따른 편향으로 나타난다. 우리가 사용하는 대부분의 데이터들은 현실 지향적이다. 이미지넷(ImageNet)의 데이터와 라벨을 기반으로 사물을 인식하는 모형을 만들어 보자[*28]. 과거 사물에 대한 데이터가 부족하여 인식하지 못하는 문제를 쉽게 재현할 수 있다. 아이폰은 인식하지만 키보드 달린 블랙베리는 인식하지 못한다. 나온 지 15년밖에 되지 않은 PDA도 인식하지 못한다. 오디오 컴포넌트는 인식하지만 턴테이블은 인식하지 못한다. 무작위로 웹에서 수집한 이미지일 경우에도 동일한 문제가 있다. 단위 시간당 데이터의 양은 카테고리를 막론하고 기하급수적으로 증가하고 있다.

이 문제는 통시적일 뿐 아니라 공시적인 문제이기도 하다. 전세계의 IT 발전 정도는 균일하지 않다. 지역에 따른 데이터 밀도 차가 발생한다. 전산화가 늦거나 사용 인구가 적은 지역들은 데이터 확보가 늦다. 자연어 인식과 자율 주행 등이 대표적인 예다.

[그림 4] InceptionV4+ImageNet에 2002년형 PDA를 분류했을 때 결과 AI는 PDA를 인식하지 못한다.



동적 평형 시스템에서 생성되는 데이터: 대상 시스템의 진화 문제

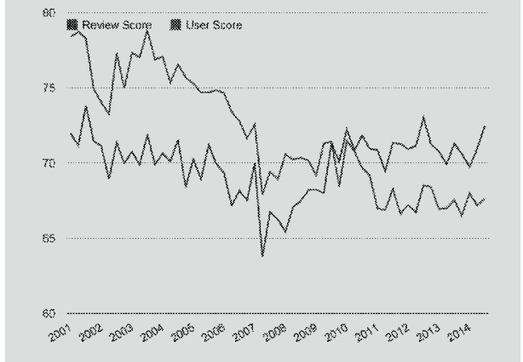
인공 투자자는 주식 투자자들의 꿈이다. 기계 학습은 알고리듬 매매에 오래전부터 사용되어 왔다. 인공 신경망의 투자 응용도 비교적 오래전부터 적용된 분야이다. 기계 학습이 패턴 인식에 강한 특성이 있기 때문이다. 그런데 지속적으로 엄청난 돈을 벌어들인 단일 모형은 등장하지 않았다[*29].

모든 기계 학습의 기본 가정은 훈련 입력과 추론 입력의 통계적 특성이 동일하다는 것이다(통계적 관점에서 정적 평형 상태의 시스템을 가정한다[*30]). 그러나 통계 및 회귀 모형들에서 정확도를 높이기 위하여 인공 신경망 모형을 도입한 경우들의 상당수는 동적 평형 시스템이다[*31]. 동적 평형 시스템은 동일한 시스템일지라도 시간에 따라 통계적 특성이 변한다. 그러므로 모형의 추론 결과가 맞지 않는 경우가 쉽게 발생한다. 주식시장은 대표적인 동적 평형 상태의 시스템이다.

주가를 예측하는 간단한 기계 학습 모형을 만드는 과정을 가정하자. 최근 10년간의 코스피(KOSPI) 데이터를 다운로드 받고, 과거 8년의 데이터로 마지막 2년의 주가를 예측하는 RNN 기반의 모형을 설계할 수 있을 것이다. 조금 노력한다면 회귀 분석 모형에 비해 평균적으로 조금 높은 예측 정확도를 얻을 수 있을 것이다. 그런데 기간 수익률은 평균 수익률과 차이가 나지 않는다. 보통 예측 향상에 의해 발생하는 상대 이윤을 예측이 실패한 경우의 더 커진 손해로 인해 잃기 때문이다. 실무 단계가 되면 더 심각한 잠재적인 문제들도 있다. 신경망 모형에서 가끔 발생하는 과적합이 동적 시스템의 상태 변화와 만날 경우 주식 투자 모형에서 큰 손해로 이어질 수 있다[*32].

시간에 따라 변하는 시스템으로 조금 더 재미있는 시도를 해 보자. 게임에 대한 각 매체의 평점을 수합하여 평균 점수를 내는 메타크리틱(Metacritic)[*33]이라는 사이트가 있다. 게임 표지 이미지를 바탕으로 게임의 성공 확률을 예측하는 모형을 만들 수 있을까? 모형을 훈련시키기 전에 이 사이트의 시간에 따른 게임 평점 분포를 살펴보자. 분기별, 연도별로 큰 변화가 있다.

[그림 5] Metacritic 의 분기별 평점 변화[*34]. 이걸 표준화하는 것이 가능할까



이 데이터를 정규화할 수 있을까? 가장 쉽게 떠올릴 수 있는 방법은 정규 분포화 이다. 각 분기별 평균을 기준으로 정규 분포가 되도록 게임 평점을 스케일할 수 있을 것이다[*35]. 그런데 좀 다르게 생각해 보자. 이 데이터가 정규 분포화 되어야 하는 데이터일까? 연도별 관점에서 보면 평균 평점이 낮은 해는 정말로 게임들이 재미가 없는

해였을 수도 있다. 또는 평점이 높은 해에 재미있는 게임이 몰려 나왔을 수도 있다. 분기별 관점에서 보면, 연말 시즌 전후에 게임들이 몰려 나오므로 그 전후가 최고 점수가 더 높고 분산은 더 큰 구간일 것이다. 그러면 이 데이터를 정규 분포에 끼워 맞추는 것은 잘못된 접근 방법일 것이다. 적절한 방법을 떠올릴 수 있는가?*36

이러한 문제를 해결하기 위한 일반적인 접근은 실시간 훈련을 적용하는 것이다. 그러나 신경망 모형을 실시간으로 훈련하는 것은 다양한 이유로 사실상 불가능하다. 데이터 공급기를 실시간 모형에 붙이는 과정은 데이터의 크기가 문제가 된다. 과적합을 막기 위해 탈락(dropout)을 적용할 경우, 탈락을 실행하는 주기마다 추론 정확도가 영향을 받는다. 따라서 기계 학습 모형을 실사용하는 경우 훈련은 연속적이 아니라 주기적으로 실행하는 것이 일반적이다*37. 이는 공시적으로는 정적 평형을 유지하지만 통시적으로는 동적 평형 상태에 있는 시스템에 적절한 방법이다.

모형 학습 시의 각인효과 : 데이터 라벨/카테고리별 밀도차

많은 신경망 모형들은 기존 방법론으로는 잘 되지 않는 복잡하고 유사해 보이는 데이터들을 분류하거나 묶기 위해 훈련된다. 신경망 모형의 약점 중 하나는 과적합이다*38. 과적합을 가장 쉽게 유도하는 방법은 특정 카테고리에 치우친 훈련 데이터를 사용하는 것이다.

최근의 사진 관리를 위한 다양한 도구들에는 기계 학습 모형들이 들어 있다. 아이폰 사용자는 사진앱(Photos)를 쓸 수 있고, 안드로이드 사용자는 구글 포토(Google Photo)를 쓸 수 있다. 둘 모두 faces라는 끝내주는 기능이 있다. 아이폰에서는 '사람들'로 부르고, 구글에서는 '인물' 이라고 부른다. 기계 학습 모형을 사용하여 사진에서 얼굴을 찾아내고, 누구인지 인덱싱하는 기능이다. 아직 많이 써 보지 않은 사용자라면 재미있는 실험을 할 기회가 있다. 아이폰 사진앱이나 구글 포토를 열어 보자. 자동으로 찾지 못한 내 사진을 찾아 수동으로 라벨을 붙여 볼 수 있다. 학습 모형이 추천한 내 후보 사진들을 보고, 맞음/틀림 입력을 주어 훈련도를 높일 수 있다.

나르시스트가 아니더라도 자신의 사진 앨범엔 본인 사진이 많기 마련이다. 한참 훈련시키다 보면 의도적으로 과적합 상태를 만들 수 있다. 어느 정도 굴리고 나면 사진앱이 보기엔 여자 친구도 나 같고, 옆집 아저씨도 나 같고, 지나가던 사람 닮은 고양이 얼굴도 나 아니냐고 물어볼 것이다.

분류 모형의 훈련을 위해 수집하는 데이터들 중 인위적인 분류를 거치지 않은 데이터의 카테고리별 분포는 일반적으로 멱함수 분포를 따른다*39. 그러므로 임의의 데이터를 임의로 수집할 경우 라벨 분포는 반드시 치우치게 된다. 간단한 실험을 해 보자. 기계 학습의 "Hello World"라 불리는 MNIST 손글씨 분류 훈련 데이터에서, 일부러 몇몇 숫자들의 샘플 비율을 낮춘 후 훈련에 사용해 보자. 무작위일 때와 차이 나는 결과를 얻을 수 있다*40.

데이터 편향성은 신경망 기반의 모형이 '편견'을 갖게 되는 가장 큰 원인이다. 실제 세계의 데이터로 훈련된 모형은 추론 과정을 통해 역으로 실제 세계에 영향을 미치기도 한다. 구글의 다양성 리포트*41에서 포용적 기술(inclusive technology)을 제시하며 발표한 실례들이 있다. 스마트폰 카메라 앱에서 흑인이 피사체에 포함된 경우 얼굴 탐색이 제대로 이루어지지 않거나 톤이 망가지는 예나, 보편적인 신발 데이터를 훈련시켰는데 하이힐의 비중이 적어 하이힐은 잘 찾아내지 못하는 경우 등이다*42.

강제로 라벨당 데이터의 비율을 맞추는 방법이 가장 쉬운 해결책이다. 이 해결책은 바로 다른 문제에 직면한다. 비중이 적은 라벨의 샘플 수에 다른 데이터의 샘플 수를 맞추다 보니 사용 가능한 데이터가 너무 적어지는 문제이다*43. 이 문제를 우회하기 위해서는 다단계 분류자를 이용하여 가장 큰 샘플 수를 갖는 분류부터 차례차례 분류하고, 제외한 나머지 데이터들을 계속 반복 분류하는 방법이 있다. 이 방법은 분류 항목들에 계층 구조가 있을 경우는 잘 동작하지만, 그렇지 않은 경우에는 사용할 수 없는 문제가 있다*44.

나가기

앞에서 재미있게 알아보았듯이*45 신경망 모형을 훈련할 경우 모형의 구조만큼이나 중요한 것은 훈련 데이터이다. 훌륭하게 전처리된 훈련 데이터는 모형 구조의 최적화 및 간략화에 큰 영향을 끼치며, 훈련에 들어가는 엄청난 자원을 절약하도록 돕는다.

데이터 전처리 과정에는 해당 분야에 대한 전문적인 지식 및 통찰이 필수적이다. 무엇을 추론할 것인지가 명확한 경우, 필요한 특징이 함께 명확해지는 경우가 대부분이다. 모형 설계자는 데이터를 기반으로 어떤 특징을 사용할지를 결정한다. 그 후 특징들의 상호 관계를 분석하여 필요한 특징을 선택하거나*46, 원하는 특징이 없는 경우 특징들을 결합하여 합성 특징을 만든다. 유의미한 특징을 정의하는 과정에서 해당 분야에 대한 지식이 매우 중요하다. 모형 훈련 과정에 사용할 데이터 표본을 대상으로 다양한 통계 분석을 실시하고, 그에 따라 적절한 특징을 선택하기 위해 해당 분야의 지식이 필요하기 때문이다.

신경망 모형 설계의 초기 접근에 필요한 기술적인 난이도는 다양한 오픈소스 툴킷들과 라이브러리에 힘입어 지속적으로 낮아졌다. 2017년 말이 되면(석 달 후임에도 불구하고) 현재보다 더 쉬워질 것이다. 텐서플로우(TensorFlow)는 차차기 버전에서 공개할 새로운 명령형 프로그래밍 모드를 준비하고 있다. 파이토치(PyTorch)는 성능상의 단점에도 불구하고 코딩 편의성과 RNN에서의 상대적 성능 이점을 내세워 사용자층을 넓혀 가고 있다. 아마존의 엠엑스넷(MxNet)와 마이크로소프트의 인지툴킷(Congnitive Toolkit, CNTK)도 넓은 호환 언어 및 뛰어난 성능을 바탕으로 케라스(Keras)와 짝을 지어 급격하게 활용 예를 늘려가는 중이다.

이에 따라 앞으로의 신경망 모형 개발 과정에는 같은 특정 분야 전문가*47의 역할이 갈수록 중요해질 것이다. 신경망 전문가가 특정 분야의 전문 지식을 쌓는 것보다 그 분야의 전문가가 신경망 작성 및 설계 기술을 배우는 것이 곧 더 쉬워질 것이기 때문이다*48. 이러한 변화는 신경망 훈련 데이터 전처리에 활용할 수 있는 여러 도구들의 등장에서도 읽을 수 있다. 최근에 아마존에서 대용량 데이터의 전처리를 돕는 서비스로 글루(Glue)*49를 출시하였다. 페어(People + AI Research Initiative, PAIR)*50의 결과로 2017년 7월에 공개한 구글 패싯(Facets)*51의 경우, 데이터 시각화를 통해 통계 분석과 특징 추출을 직관적으로 돕는 도구로 주목할 만하다.

쉬워 보이지만 막상 모형이 잘 동작하지 않는 경우 짚어 보아야 하는 다양한 부분들 중 데이터에 관련된 부분들을 다루어 보았다. 기계 학습 보급의 초입에서 만나게 될 수많은 장밋빛 전망들이 정작 내 손에서는 재현되지 않을 때, 마치 신경망 분야에 사기당한 것 같을 때마다 한번 생각해 보자.

"지금 내가 내 모형에 밥 대신 다른 걸 먹이고 있는 것이 아닐까?"

*1 참고 | 대표적인 문제로 과적합 , 가중치 포화 문제 등이 있다. 과적합을 예로 들어 보자. 인공 신경망의 경우 (시간 축을 무시할 경우) 파라미터 공간이 비유클리드 공간이므로 데이터의 특징에 상관 없이 그 데이터를 가장 잘 분류하도록 훈련할 수 있다. 그렇지만 그 반대급부로 훈련 데이터 및 횟수가 적은 경우 오차 민감도가 너무 올라가서 원하는 결과와 거리가 먼 분류면을 만드는 문제가 있다. 이 문제는, 일부의 신경망 연결을 무작위로 해지하여 과적합 상태에서 강제로 나오게 하고, 대량의 데이터를 이용하여 훈련하거나 (강화학습의 경우) 시뮬레이터로 훈련 횟수를 늘려 해결할 수 있게 되었다. *2 참고 | 보통 '딥러닝' 이라고 부르며, 학술적으로는 10여 개 이상의 은닉층을 사용하는 경우를 뜻한다. *3 참고 | end-to-end 모델. 전처리 및 후처리와 최적화 과정 없이 데이터를 입력 특징으로 주면 내부에서 모든 처리를 수행하여 최종적으로 우리가 원하는 출력이 나오는 모형. *4 참고 | 선형적으로 증가하지 않고 지수적으로 증가한다. 단, 복잡도가 거듭제곱 꼴로 증가하지는 않는다. 이는 일반적인 인공 신경망의 연결 구조가, 모든 뉴런들이 연결된 것(all-to-all)이 아니라 각 층의 뉴런이 다음 층의 뉴런들과만 연결되어 있는 다중분할(multipartite) 구조이기 때문이다(드물지만 예외도 있다). *5 참고 | 수학적으로 인공 신경망의 훈련 과정은 마르코브 과정이므로 분산 처리에 적합한 모형은 아니다. 수학적 엄밀성이 필요하지 않은 응용 및 수치적인 접근 차원에서 분산 처리를 이용한 훈련 가속을 목적으로 미니 배치 등을 사용하고 있다. *6 참고 | 단일 작업을 처리하기 위해 하나 이상의 기계 학습 모델을 직렬 또는 병렬로 연결한 모델 그룹을 만들어 문제를 해결하는 모형 및 방법론. *7 참고 | 극댓값(Local maximum)이 두 개인(maxima)인 분포를 말한다. 여러 개인 경우는 멀티 모달(Multi-modal)이라고 부른다. *8 참고 | 크게 두가지 이유가 있다. 신경망 모형의 출력 노드 수가 적은 경우 학습 데이터 카테고리의 데이터 비율에 크게 영향을 받는다. (여기서는 두 개뿐이다.) 또한 새로운 데이터에 새로운 라벨이 붙은 경우가 아니라, 동일한 데이터의 점이값들에 라벨을 다르게 붙여 훈련한 경우이므로 모형이 표현하는 상태 공간이 완전히 다르게 정의된다. 게다가 신경망 모형을 쓰는 경우라면 이미 데이터가 성기게 분포하고 있어 상태 공간을 충분히 설명하지 못하는 상황일 것이다. 쓰시마섬이나 독도에 어떤 국가 라벨을 붙이느냐에 따라 영해가 어떻게 바뀌는지 상상해 보자. *9 참고 | https://medium.freecodecamp.org/whose-reviews-should-you-trust-imdb-rotten-tomatoes-metacritic-or-fandango-7d1010c6cf19 *10 참고 | https://play.watcha.net *11 참고 | 두 데이터의 평점 분포를 확인하면 차이를 쉽게 알 수 있다. *12 참고 | 메타 사이트는 선형 스케일로 데이터를 맞추는 대표적인 경우이다. 로튼 토마토 (http://rottentomatoes.com) 서비스 등이 대표적인 메타 평점 사이트이다. 이 서비스는 영화 평론 사이트들의 평점을 강제로 100점 기준으로 스케일하고 평균 평점을 내는 사이트이다. 이러한 메타 평점 사이트들이 내재하고 있는 통계적 문제점에 대한 많은 분석 결과들이 있다. *13 참고 | https://www.theverge.com/2017/3/16/14952434/netflix-five-star-ratings-going-away-thumbs-up-down *14 참고 | https://youtube.googleblog.com/2009/09/five-stars-dominate-ratings.html *15 참고 | 23앤드미(23andMe)처럼 전체 DNA 배열을 분석하는 대신 일부 특정 질환의 예측을 위한 스타트업을 창업한다고 가정해보자. *16 참고 | 유전자 미세배열(Gene Microarray) *17 참고 | 유전자 발현분석(Gene Expression Profiling) 작업 과정을 단순화한 설명이다. (직접 해 볼 수도 있다. 온라인에서 연구용 목적으로 공개되어 있는 유전자칩(GeneChip) 데이터들이 많다. PLEXdb (http://www.plexdb.org/modules/PD_general/tools.php) 등을 참조하라) 요새는 이럴 필요 없이 (돈이 있으면) 혈액을 이용해 유전정보 전체를 시퀀싱하고 통계 처리해서 바로 알아낼 수 있는 시대이다. *18 참고 | 과학의 근본 원리인 동일 현상에 대한 동일 결과(실험의 확증성)에 반하는 것처럼 보인다. 그러나 측정 도구도 측정 대상계의 일부이기 때문에 어쩔 수 없이 나타나는 현상이다. 바이오 분야(를 포함한 실험 과학 전반)에서는 비일비재하다. 동일 브랜드의 동일한 기기에서는 동일하거나 비슷한 결과가 나오므로 한정적 상황에서는 실험의 확증성을 위반하지 않는다고 할 수 있다. (이렇지 않은 기기는 팔 수가 없을 것이다) 이러한 이유로 논문이나 연구 문서의 경우 반드시 실험에 사용한 기기를 명시하고 있다. *19 참고 | 이러한 데이터들을 추가적인 통계 처리()를 이용해 동일한 데이터 세트로 표준화하려는 노력도 지속적으로 이루어지고 있다. *20 참고 | 원시 데이터(Raw data): 기기에서 바로 측정한, 가공을 거치지 않은 데이터. *21 참고 | 다양한 이유로 데이터가 불안정하다. *22 참고 | 생물체에서 정량적인 데이터가 제대로 나오는 경우는 드물다. 그래서 통계 처리가 매우 중요하다. *23 참고 | 원시 데이터를 보면 사람이 얼마나 많은 단어를 생략하고 말할 수 있는지 깨닫게 될 것이다. *24 참고 | 기기 공급사의 소프트웨어 업그레이드에 의해 데이터 후처리 과정이 데이터 수집 중간에 변경될 수 있는 가능성을 막기 위한 방법이다. 다양한 파이프라인 소프트웨어가 있음에도 직접 작성을 권장하는 이유이다. 또한 전처리 파이프라인을 따로 둘 경우 기계 학습 모형에 사용할 특징을 바꿀 경우 유연하게 대응할 수 있다. 머지않은 미래에는 종단간 모형에 원시 데이터를 바로 집어넣는 모형도 할 수 있을 것이다. *25 참고 | 최근의 시도로는 2017년 8월 아마존의 에코룩(Echo Look)(https://www.amazon.com/Echo-Hands-Free-Camera-Style-Assistant/dp/B0186JAEWK)이 기계 학습 모형을 이용하여 사용자의 취향 및 트렌드에 따른 맞춤형 옷을 주문 제작하는 서비스를 테스트하고 있다. *26 참고 | 연구용 목적으로는 DeepFashion Dataset (http://mmlab.ie.cuhk.edu.hk/projects/DeepFashion.html) 등으로 시작할 수 있다. *27 참고 | 1993년~2005년의 NYSE 데이터로 연구를 했을 때 기록해 둔 용량이다. 연단위로 그래프를 그리면 전형적인 지수 증가 추세를 보인다. 이 경향이 여전하다면 아마도 2016년 이후에 생성된 데이터의 양이 2016년 이전에 생성된 모든 데이터의 합보다 많을 것이다. *28 참고 | 구글의 InceptionV4 의 경우 학습이 된 신경망+라벨을 다운로드할 수 있다. https://github.com/tensorflow/models/tree/master/official/resnet *29 참고 | 물론 여러 투자 모형을 결합한 멀티모달 그룹 모형의 경우 이미 여러 투자회사 및 금융기관에서 사용하고 있다. *30 참고 | 동일한 (역학에서의 무게중심과는 콘셉트만 같은 개념인) 무게중심, 입력 데이터간의 독립 항등 분포 (i.i.d., independent and identically distributed), 동일한 n차 모멘트 등. *31 참고 | 정적 평형 상태의 시스템에서는 데이터의 특징 공간이 너무 크지 않고 특징 분포가 복잡하지 않으면 대부분의 회귀 모형이 어느 정도 이상의 결과를 내놓는다(그래서 신경망 모형까지 도입할 필요가 없다). *32 참고 | 인공 투자 시스템의 오류로 인하여 발생한 여러 (알려지거나 알려지지 않은) 사건이 있다. 알려진 사건 중 유명한 사건은 나이트 캐피탈(Knight Capital)이 2012년에 4억 4천만 달러를 30분 동안 날린 사건이다.http://www.businessinsider.com/market-trading-issues-knight-capital-tanking-2012-8 , http://www.businessinsider.com/knight-capital-is-facing-a-440-million-loss-after-yesterdays-trading-glitch-2012-8 을 참고. *33 참고 | 참고: http://www.metacritic.com/ *34 참고 | https://www.polygon.com/2014/10/28/7083373/look-at-this-chart-of-average-metacritic-scores-what-happened-in-2007 *35 참고 | 이 방식으로 계산된 대표적인 값은 수학능력시험의 표준점수이다. *36 참고 | 시간 축을 x로, 평점을 y로 놓은 시계열 데이터를 만들어 탈경향변(Detrended Fluctuation Analysis, DFA)을 돌리고, 시기에 따른 영향이 어느 주기로 나타나는지 파악하는 것으로 시작해 보라. *37 참고 | 실제 해 보면 이 경우도 문제가 생기는데, 모형이 오래된 것일수록 훈련의 이득이 거의 없어진다. 상황에 따라 다양한 해결 방식이 있을 것이다. *38 참고 | 일정 주기로 신경망의 연결을 무작위로 제거하는 탈락(dropout)이 과적합을 막기 위하여 널리 쓰인다. 야매 같아 보이는 탈락이 이렇게 널리 오래 쓰일 줄은 아마 아무도 몰랐을 것이다. (심지어 얼마 전에는 두뇌에서도 비슷한 현상이 관찰되었다.) *39 참고 | 특별한 이유가 있는 것이 아니라 무작위 선택의 누적에 따라 통계적으로 나타나는 자연의 특성이다. *40 참고 | 그런데 MNIST로는 티가 크게 나지 않는다. MNIST 데이터는 픽셀 하나를 숫자 하나 판별하는 기준으로 쓸 수 있을 정도로 정형화된 데이터이기 때문이다. fashion-MNIST 데이터 (https://research.zalando.com/welcome/mission/research-projects/fashion-mnist/) 에서는 카테고리 편향 문제를 비교적 뚜렷하게 실험할 수 있다. *41 참고 | https://diversity.google/ *42 참고 | 사진에서 물리학자들을 찾아내는 비유를 들어 모든 물리학자들이 남성이었기 때문에 마리 퀴리를 찾아내지 못하는 예를 들었다. (이는 동적 평형 시스템이 데이터에 끼치는 영향의 일례로 들 수도 있을 것이다.) GDD 유럽 2017(Google Developer Day Europe 2017)에서 편향에 대해 다룬 동영상을 참고하라. https://youtu.be/ZgaQn9coYfU?t=27m23s *43 참고 | 물론 원 데이터가 엄청나게 큰 경우는 상관 없다. *44 참고 | 계층 구조를 정의할 수 있는 데이터의 예로 동물 분류 데이터. 계층 구조가 없는 데이터의 예는 손글씨 데이터. *45 참고 | 글 말미라 하는 이야기이지만 당시 저 주제들이 해결해야 하는 주제였을 때는 재미있진 않았다. *46 참고 | 특징을 선택하는 기존은 일반적으로는 피어슨 상관관계 같은 상호간의 연관성이 가장 낮은 값들인 동시에, 결과 라벨을 가장 잘 설명하는 특징들이다. 그러한 특징이 없는 경우, 수학적으로 변형된 특징(예: 제곱, 제곱근, 절대값, 초월함수값)들로 테스트하거나, 또는 두 특징을 합성하여 (예: 특징들의 곱, 특징들의 합, 특징들의 차) 사용하기도 한다. *47 참고 | 도메인 전문가(Domain specialists) *48 참고 | 그렇더라도 머신러닝 전공자들은 걱정하지 말자. 할 일이 차고 넘친다. 엑셀이 보급되어도 데이터 과학자들은 잘 살아남았다. *49 참고 | https://aws.amazon.com/ko/glue/ *50 참고 | https://ai.google/pair *51 참고 | https://pair-code.github.io/facets/ (약간의 광고성 링크이지만) 구글이 직접 소개하는 포스트를 참고하라. https://developers-kr.googleblog.com/2017/08/facets-open-source-visualization-tool.html