KAKAO AI REPORT

Vol.07

2017.10

```
AI CODE
def getCTRWithItems(targetSegments: Seq[String],
                    timeSeries: TimeSeriesData,
                    default: Double)
: Future[Map[SegmentT, Double]] = {

  val segmentPCTRsResult = super.getCTR(targetSegments, timeSeries, default)
  val segmentRawsResult = super.getRaws(targetSegments, timeSeries)
  val contentFeature = getContentFeature(timeSeries.itemId)

  val result = for {
    mediaDurations <- MediaArticlePerusal.score(pctrCode,
                                                timeSeries.channel,
                                                Seq("u"),
                                                timeSeries.itemId,
                                                contentFeature)
  } yield {
    targetSegments.map { segmentId: String =>
      val mediaArticlePerusalRateScore = mediaDurations.getOrElse("u", DefaultValue)
      val ensembleScore: Double = ensemble(default,
                                           segmentPCTRsResult,
                                           segmentId,
                                           mediaArticlePerusalRateScore)
      segmentId->ensembleScore
    }.toMap
  }

  result
}
```

카카오에서 매월 발행하는 AI 리포트 입니다.

kakao

KAKAO AI REPORT

Vol.07

발행일 | 2017년 10월 31일
발행처 | (주)카카오
발행인 | 카카오 정책지원파트
편집 | 김대원, 문정빈, 양원철, 양현서, 정수헌
디자인 | 허진아
메일 | kakaoaireport@kakaocorp.com

COVER

카카오 AI 리포트의 표지에선 AI와 관련된 의미 있는 코드들을 매월 소개하고 있습니다.

Vol.07 코드 | 성인재 tevin.sung@kakaocorp.com
표지 코드는 모바일 다음 메인 화면의 뉴스 기사와 콘텐츠 추천에 적용되는 카카오I(아이)의 추천 엔진 알고리즘 일부를 발췌한 것이다.

contents

카카오 AI 리포트 7호를 내며

이번 호는 9/10월 합본호 입니다. 이번 호에서는 카카오미니(Kakao Mini)의 이야기를 우선 담았습니다. 카카오미니는 텍스트가 아닌 사람의 음성에 반응하는 인공지능(artificial intelligence, AI) 기기입니다. 이 새로운 소통 방식을 처음 접하시는 분들은 "내 목소리를 어떻게 인식하지?" 라는 궁금증이 드실 겁니다. 그래서, 말하는 사람의 음성이 어떻게 인식되는 지에 대한 기술적 설명을 앞 부분에 넣었습니다. 글은 카카오미니의 개발을 맡고 계신 김명재 님이 작성해 주셨습니다.

음성(voice)이 새로운 플랫폼으로의 접점으로 부상하게 된 상황과 그 변화에 대해서 카카오미니의 실무를 총괄하는 이석영 님이 설명해 주셨습니다. 이석영 님이 정리하신 카카오미니의 의미는 다음과 같습니다. "스마트폰이 등장하고 10년 동안 세상이 더 편리해진 것처럼, 음성 대화 인터페이스와 새로운 가정용 스마트 디바이스의 출현은 앞으로 오랜 시간에 걸쳐 삶의 많은 부분을 변화 시키고 새로운 가치를 만들어 낼 것이다."

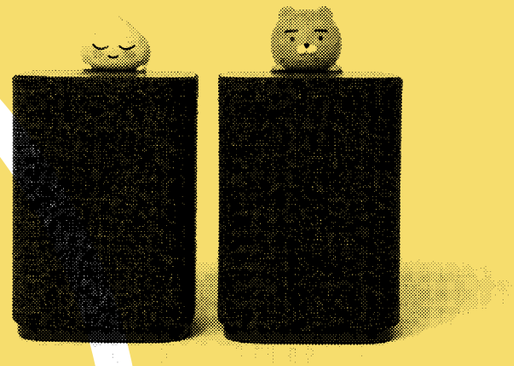
카카오미니 이야기에 이어 카카오I(아이) 추천 엔진의 최근 진화 상황을 정리했습니다. 뉴스에 적용된 추천 엔진의 알고리즘에는 이용자의 콘텐츠 소비 빈도 외에 체류 시간 등 소비 형태 및 양상까지도 담겼습니다. 딥러닝의 성과와 데이터 간 상관성에 대한 치열한 고민을 담은 신정규 님의 글은 주요 인공지능 커뮤니티에서 빠르게 공유될 것으로 예상됩니다. 카카오브레인의 이수경 님은 '알파고 중의 알파고'로 불리는 알파고 제로(AlphaGo Zero)와 기존 알파고 간의 차이를 간단 명료하게 정리해 주셨습니다.

카카오AI 리포트로 최신 연구 동향을 엿보고 싶으신 독자 분들을 위해 주요 대학의 인공지능 연구실로부터 꼭 봐야 하는 논문을 추천받았습니다. 데이터, 의료, 공학까지 다양한 전공분야의 연구실에서 글을 주셨습니다. 카카오브레인의 안다비 님과 천영재 님은 올해 열린 ICML과 CVPR에 다녀오신 후, 최신 AI 논문이 발표되는 뜨거운 현장의 분위기를 전달해 주셨습니다.

전설의 게임인 슈퍼마리오에 강화학습을 적용한 이야기를 다룬 카카오 데이터 엔지니어 송호연 님의 글은 이번 호를 시작으로 12월호까지 3번에 걸쳐 연재될 예정입니다. AI에서 가장 각광받고 있는 분야 중 하나인 generative adversarial networks, 일명 GAN의 주요 개념을 카이스트의 유재준 님이 이번 호와 다음 호에 걸쳐서 설명해 드립니다. 11월과 12월에 열리는 국내외 AI 컨퍼런스 소개글을 이번 호의 마지막 콘텐츠로 담았습니다.

2017년 10월 31일
카카오 정책지원파트 드림

A special edition



kakaomini

다음 달에 정식 판매 될 카카오미니를 떠올리면, 아마도 "카카오미니가 내 말을 어떻게 인식하지?"라는 생각이 드실 겁니다. "똑똑한 인공지능이 탑재되어서 그렇겠지!"라는 막연한 답변 말고, 기술적인 설명이 궁금한 분들이 많으실 겁니다. 키보드를 손으로 치거나, 터치 스크린을 누르는 물리적 행위로 기기에 의사를 전달하는 구조에서 사람과 기기 간 소통 방식은 명확합니다. 허나, 음성이란 보이지 않는 형체가 사람의 메시지를 실어 나르는 구조는 대다수에게 생경할 겁니다. 카카오미니가 소비자 분들에게 신선하게 여겨질 이유 역시도 그 때문이라고 생각됩니다. 이 궁금증에 대해, 카카오미니의 음성인식 기술 개발을 담당하고 있는 김명재 님이 명쾌하게 설명해 주셨습니다. 기술 이야기에 앞서, 카카오미니 프로젝트 총괄을 맡고 있는 이석영 님이 터치(touch)에서 보이스(voice) 중심 플랫폼으로의 기술 변화 맥락을 소상하게 설명해 주셨습니다.

세상을 바꿀 변화의 시작, 음성 인터페이스와 스마트 스피커

지금으로부터 10년 전인 2007년, 미국 샌프란시스코에서 중요한 세상의 변화가 시작되었다. "오늘 세상을 바꿀 세 가지 디바이스(device)를 선보일 것입니다. 와이드 스크린에 터치 인터페이스로 동작하는 아이팟(iPod), 혁신적인 휴대 전화, 그리고 획기적인 인터넷 커뮤니케이터. 아이팟, 전화기, 인터넷 커뮤니케이터… 눈치 채셨나요? 네, 이건 세 가지 디바이스가 아닙니다. 단 하나의 디바이스, 바로 아이폰(iPhone)입니다."

2007년 맥월드(MacWorld) 키노트에서 스티브 잡스가 얘기한 것처럼, 애플은 휴대폰을 재창조했고 세상을 완전히 변화시켰다. 이 '작은 디바이스' 하나만 있으면 언제 어디서나 전화와 음악듣기는 물론, 인터넷과 연결된 수많은 서비스들을 편리하게 이용할 수 있게 됐다.

글 | 이석영 zodiac.lee@kakaocorp.com

현재 카카오에서 카카오의 인공지능 플랫폼인 카카오I와 이를 활용한 카카오의 첫 번째 스마트 스피커 카카오미니 프로젝트를 총괄하고 있다. 닷컴 버블 전인 1999년 부터 IT 서비스 기획을 계속 해오고 있으며, 웹서비스와 스마트폰이 세상을 크게 변화시켰던 현장의 중심인 네이버와 카카오에서 일을 해왔다. AI시대를 맞아, 사람들의 삶을 한번 더 진화시키기 위한 방법을 찾는것이 가장 큰 관심사이다.

스마트폰, 세상을 바꿔버린 슈퍼 디바이스

무엇보다 스티브 잡스가 아이폰을 소개하며 강조한 것처럼, 스마트폰은 다른 휴대용 기기를 통합했다. 오늘날 이 특징은 너무나 당연하게 여겨지고 있지만, 2000년대 초반만 해도 그렇지 않았다. 외출할 때 마다 사람들은 가방에 휴대폰, MP3 플레이어, 디지털 카메라, 휴대용 게임기, PMP(DMB), 전자사전 등의 기기를 넣고 다녀야 했고, 필요할 때 마다 그 중 하나를 꺼내어 이용하곤 했다.

단순히 생활 양식이 바뀐게 아니라, 앞서 언급된 디바이스들은 실제로 세상에서 자취를 감추거나 다른 형태로 바뀌었다. 한 때 필수품 취급으로 받았던 휴대용 MP3 플레이어는 거의 사라졌고, 디지털 카메라 시장은 고급형 모델 중심으로 재편되었다. 스마트폰은 이제 세상에서 가장 많은 사람들이 사용하는 휴대용 게임기가 되었고, 동영상앱을 실행하여 스마트폰으로 영화 한 편을 보는 시대가 되었다.

스마트폰의 등장으로 세상에서 자취를 감춘 디바이스는 휴대용 기기에 국한되지 않는다. 자동차에서는 거치용 내비게이션이 점점 사라지고 있고, 집에서는 종합 콤포넌트라고 불리던 오디오 데크가 사라졌다. 집에 유선전화를 개통하는 경우가 크게 줄었으며, 방마다 하나씩 있었던 탁상용 알람 시계 역시 어느 순간 찾아볼 수 없게 되었다.

이 모든 것을 사용하기 위한 단 하나의 인터페이스, '터치'

모든 개별 디바이스의 기능을 하나로 통합한 스마트폰은 "터치 인터페이스"라는 혁신적인 기술이 있었기에 존재 가능했다. 아이폰 이전에도 스마트폰이라고 할 만한 디바이스가 있었으나, 대부분 쿼티(QWERTY) 키보드를 장착하거나, 전용 스타일러스(stylus) 펜을 이용하는 형태였다. 스티브 잡스는 이를 대단히 못마땅 하게 생각했다. "신(神)은 우리에게 이미 스타일러스를 주셨어. 그것도 열 개나." 스티브 잡스가 아이폰 개발팀에게 자기 손을 흔들어 보이며, 자연스러운 터치 인터페이스의 개발이 가장 중요하다고 말한 이야기는 널리 알려진 일화다. 2007년 키노트에서 잡스는 아이폰의 가장 중요한 핵심 기능으로 손가락만으로 동작 가능한 '멀티 터치 인터페이스(multi touch interface)'를 첫 번째로 소개했다. 기존 쿼티 키보드와 스타일러스 펜이 가진 끔찍한 사용성을 함께 언급하면서.

실제로 터치 인터페이스는 매우 쉽고 훌륭하다. 미세한 손가락 움직임에도 반응하는 정전식 디스플레이 장치는 기존 입력 인터페이스 장치들(키패드, 마우스, 포인팅 장치 및 각종 버튼들)이 가지는 조작성의 한계를 사실상 완전히 없앨 수 있기 때문에, 스마트폰에 담겨있는 서비스와 기능들을 거의 무한대에 가까운 방식으로 이용할 수 있게 해준다. 게다가 손가락으로 화면에 표시된 무언가를 눌러 반응을 보는 것은 학습비용이 매우 낮을 뿐만 아니라 자연스러운 형태의 인터페이스이다. 두 세살짜리 아이가 아이폰을 쉽게 조작하는 모습을 보는 것은 이제 별로 놀라운 일도 아니다.

[그림 1] 터치 인터페이스

스마트폰으로의 과도한 통합이 초래한 불편함

너무 많은 기능들이 스마트폰에 담기고 이를 오직 "터치 인터페이스"로만 사용할 수 있게 되면서 불편해진 것도 있다. 대표적인 사례가 가정에서의 음악 감상이다. 과거에는 음악 감상을 위해 테이프나 CD를 데크에 넣고 플레이(Play) 버튼만 누르면 원하는 음악을 좋은 음질로 즉시 들을 수 있었다. 스마트폰과 음악 스트리밍(streaming) 서비스를 통해 언제 어디서나 다양한 노래를 들을 수 있게 되어 편리해진 '수혜'를 모든 사람이 쉽게 누릴 수 있는 것은 아니다. 모든 서비스와 산업이 스마트폰에서 소비되는 현대에, 누구나 쉽게 접할 수 있었던 음악감상을 할 수 없게 된 새로운 소외계층이 생겨났다. 아직도 적지 않은 50대 이상 장년층과 노년층에게 있어 스마트폰과 스트리밍 서비스를 통한 음악 감상은 매우 어려운 과업이다. 음악을 좋은 음질로 듣기 위해, 스마트폰을 블루투스(bluetooth) 스피커와 연결해야 하는데 이는 더더욱 어렵다.

터치 인터페이스 역시 만능은 아니다. 스마트폰은 물리적으로 화면의 크기가 제한되어 있으므로, 한 번에 제공할 수 있는 인터페이스의 정보량이 많지 않다. 그러다 보니 수많은 서비스와 기능들을 이용하기 위해서는 부득이하게 여러 번의 단계를 거쳐 서비스를 이용하도록 설계할 수 밖에 없다. 어떤 서비스를 사용하려고 해도, 보안 잠금을 해제한 후 해당 서비스 앱을 실행하여 몇 번의 터치를 거쳐야만 원하는 기능을 실행할 수 있다. 게다가 스마트폰에는 사용해야 하는 기능이 너무 많다. 터치 인터페이스 자체는 학습비용이 낮지만, 스마트폰에 익숙한 젊은 사람들 조차 앱과 스마트폰에 내재된 기능을 제대로 찾지 못한다. 휴대폰의 설정을 바꾸기 위해 여러 번의 시행착오를 겪는 젊은

사람들의 모습을 쉽게 찾아볼 수 있다.

또한 터치 인터페이스를 사용하기 위해서는 눈(시각)과 손을 필요로 하는데, 이는 태생적으로 멀티 태스킹을 할 수 없도록 만든다. 눈과 손을 온전히 스마트폰을 위해 사용해야만 원하는 기능을 얻을 수 있고 이 과정에서 다른 일들을 동시에 하는 것은 매우 어렵다. 이는 불편함을 넘어 때로는 사용자를 위험에 빠뜨린다. 보행 중이나 운전 중에 스마트폰을 사용하는 것은 대단히 위험하다. 다수의 경우가 실제로 사고로 연결되기도 한다. 스마트폰에 많은 편리한 기능이 담겨 있기 때문에, 사용자들은 앞서 말한 위험에도 불구하고 이동하며 스마트폰을 보는 경우가 많다. 이 역시 슈퍼 디바이스로써 스마트폰의 존재와 이를 터치 인터페이스로 사용해야만 하는 결합이 만들어낸 새로운 종류의 사회적 이슈이다.

가장 자연스럽고 효과적인 인터페이스, '음성 대화'

터치 인터페이스가 스마트폰과 함께 10여년간 가장 훌륭한 인터페이스가 될 수 있었던 것은 앞서 얘기한 것 처럼 자연스러움 덕분이었다. 그러나 터치 인터페이스는 '터치 스크린(touch screen)'이라는 최소한의 물리적 장치를 수반하고, 스크린 위로의 터치라는 물리적 행동을 필요로 하는 한계를 갖고 있다.
이러한 관점에서 봤을 때, 음성을 통한 대화형 인터페이스는 가장 쉽고 자연스럽게 복합적인 기능을 사용할 수 있는 방법일 것이다. 일단 음성 대화 인터페이스는 단계를 거칠 필요가 없이, 모든 서비스 이용을 한 번에 할 수 있게 해준다. 원하는 음악을 듣거나, 뉴스와 날씨 정보를 확인하거나, 알람을 맞추거나, 전화를 걸거나, 심지어 이번에 카카오미니를 통해 제공되는 기능인 카카오톡 보내기 조차도 한 번의 음성 명령으로 즉시 실행된다. 입과 귀를 사용하는 음성 대화 인터페이스는 터치 인터페이스와 달리 인터페이스 장치를 정확히 인지하지 않고도 사용할 수 있어 멀티 태스킹(multi tasking)에 훨씬 적합하다. 음성 대화 인터페이스는 앞서 얘기된 운전 중이나 보행 중에도 위험성 없이 사용할 수 있다. 아침시간 집에서 바쁘게 출근 준비를 하면서도 음성 대화 인터페이스로 날씨, 뉴스, 주가 등을 편하게 확인할 수 있다.

음성 대화를 통한 인터페이스의 가장 큰 장점은 학습비용이 터치 인터페이스보다도 낮다는 점이다. 대화라는 것은 모든 사람이 태어나서부터 배우고 이미 방법을 알고 있는 소통 방식이다. 기존 인터페이스들이 HCI(human-computer interface)라는 학문적 기반에서 발전해 왔고, 이는 사람처럼 대화를 할 수 없는 기계를 사용하기 위해 만들어진 방법임을 생각해 본다면, 음성 대화로 컴퓨터를 조작하는 행위는 궁극의 인터페이스의 한 형태라고도 볼 수 있을 것이다.

물론, 아직 음성 대화를 통한 인터페이스는 완벽하지 않다. 이것이 완전해지기 위해서는 모든 자연스러운 대화를 이해하고 응답되어야 한다. 현재의 기술로 이를 완전하게 구현하는 것은 쉽지 않다. 그러나, 음성인식과 AI기술이 빠르게 발전하고 있기 때문에 자연스러운 모든 대화를 이해하는 컴퓨터 혹은 서비스가 등장하는 것은 어쩌면 그리 멀지 않은 미래의 이야기일 수도 있다.

[그림 2] 영화 속에서 음성 대화 인터페이스가 활용되는 예시 : HER

왜 스마트폰이 아닌 스마트 스피커인가?

사실, 음성 대화 인터페이스로 터치 인터페이스가 가진 한계를 극복하고자 했던 시도는 스마트폰 제조사를 중심으로 이미 몇 년 전 부터 진행되어 왔다. 그러나 애플 시리(Siri)와 구글 어시스턴트(Google Assistant)를 통한 스마트폰 기반의 음성 대화 인터페이스는 시장에 제대로 정착하지 못했다. 음성 인터페이스는 효용성이 떨어지는 기술로 평가받으며, 훨씬 더 먼 미래에나 일상에서 사용될 수 있는 것처럼 보였다. 아마존이 에코(Echo)를 발표하여 사람들의 일상이 음성 인터페이스로 실제로 바뀔 수 있음을 보여주기 전 까지는.

아마존은 "음성 대화 인터페이스"가 가지는 특징과 가치를 제대로 이해하고 있었고, 이것이 에코가 성공을 거둘 수 있었던 가장 큰 요인일 것이다. 아마존은 전원에 상시 연결된 가정용 스피커 디바이스를 통해 알렉사(Alexa) 서비스를 제공했다. 사용자들은 알렉사를 통해 기존 스마트폰 음성 비서와는 다른 두 가지 중요한 사용자 경험을 제공할 수 있었다. 첫번째는 스피커를 24시간 음성 입력 대기 상태로 만듦으로써, 사용자가 디바이스를 사용하기 위한 별다른 준비를 하지 않아도 된다는 것이다. 사용자는 아무 때나 '알렉사'를 부르는 것만으로 서비스를 바로 이용할 수 있게 됐다. 이 과정은 서비스 이용 단계를 단 한 번으로 끝낼 수 있는 음성 인터페이스의 본질적 사용자 가치를 제대로 구현하기 위한 중요한 요소였다. 두번째는 음성 대화만으로 모든 서비스를 완전하게 이용할 수 있도록 만든 점이다. 이를 통해 사용자는 눈과 손을 자유롭게 쓸 수 있을 뿐 아니라 특별한 학습이 필요 없이 자연스러운 대화를 통해 서비스를 이용할 수 있었고, 이를통해 음성 인터페이스의 진정한 편리함을 완전하게 경험할 수 있었다.

스마트폰의 음성 비서는 알렉사가 보인 음성 인터페이스의 경험을 제대로 구현하는데 한계가 있었다. 음성 웨이크업 기능이 있지만, 보조적인 수단이었고 옵션을 꺼 두는 경우가 많아 신뢰도가 낮았다. 또한 스마트폰의 음성 비서는 터치 인터페이스의 병행 사용을 유도했는데 이로 인해 음성 대화는 스마트폰의 오롯하게 활용하는 수단이 되지 못하고, 보조 인터페이스로의 위상을 벗어나지 못하게 된다. 10년간 학습되어 온 '스마트폰 조작=터치 인터페이스 사용'이라는 명제에서 탈피해, 음성 대화로 스마트폰을 조작하는 것은 스마트폰 사용자에게는 낯선 경험으로 인식될 수밖에 없는 환경이 스마트폰의 음성 비서를 보조적 도구로 머물게 만든 이유이기도 했다.

스마트폰으로 인해 사라졌던 디바이스들의 부활

아마존이 발표하는 에코의 새로운 라인업(line up)을 보면, 스피커가 음성 대화 인터페이스 구현에 적합했기 때문에 선택된 것만은 아닌 듯 보인다. 아마존이 새로운 에코 라인업을 통해 선보이는 가장 중요한 기능 중 하나는 전화(Echo Show)와 알람 시계(Echo Spot)이다. 전화 기능과 알람 기능은 기본 에코 디바이스에서도 제공되는 기능이지만, 에코 쇼와 에코 스팟은 이 두 가지 기능을 디바이스의 형태적인 측면으로도 강조하고 있다.

아이러니하게도, 전화기와 알람시계는 에코가 대체한 '오디오 데크'와 더불어 스마트폰의 확산으로 자취를 감춘 가정용 디바이스였고 스마트폰 이전 시대에는 독립적인 기기로서 편리하게 사용되던 것들이다. 아마존의 에코 라인업은 스마트폰 시대에 없어져 버린 가정용 디바이스들을 통합하여 새로운 슈퍼 디바이스로 부활시키려는듯 보인다. 에코와 같은 스마트 스피커를 이용하면 집안에서 스마트폰을 쓰는 것보다 훨씬 편리하게 음악 감상이나 알람 설정을 할 수 있다. 가전 기기 제어와 각종 정보 확인, 커뮤니케이션, 쇼핑, 음식 주문하기도 에코를 통해 이용할 수 있다. 물론 아직은 스마트폰을 이용할 때 더 편리하게 이용할 수 있는 서비스들이 훨씬 많다. 스마트폰의 등장이 개인용 컴퓨터를 완벽하게 대체하지 않았던 것처럼, 스마트 스피커가 또 다른 슈퍼 디바이스가 된다고 해도, 스마트폰 역시 계속 사용될 것이다.

그러나 스마트 스피커는 그동안 스마트폰에게 부여된 과도한 역할 중 가정에서의 IT 서비스 사용 경험을 음성인터페이스라는 편리한 UX와 함께 많은 부분 대체할 수 있을 것이다.

변화는 이제 막 시작되었다.

스마트폰이 등장하고 10년 동안 세상이 더 편리해진것 처럼, 음성 대화 인터페이스와 새로운 가정용 스마트 디바이스의 출현은 앞으로 오랜 시간에 걸쳐 삶의 많은 부분을 변화 시키고 새로운 가치를 만들어 낼 것이다.그리고 언제나 혁신적인 생활 플랫폼을 만들어 왔던 카카오 역시, 카카오의 인공지능 플랫폼인 카카오i와 스마트 스피커 카카오미니를 시작으로 이 거대한 변화를 함께 만들어 나갈 것이다.

카카오미니는 말하는 사람을 어떻게 인식할까?

최근 음성인식의 성능이 많이 향상되면서, 음성이 친숙한 인터페이스로 자리잡고 있다. 음성에는 우리가 전하고자 하는 언어적 정보뿐만 아니라, 나이, 건강 상태, 감정 상태 등의 정보도 포함되어 있다. 또한, 음성은 각 사람의 고유한 정보를 담고 있어 이를 분석하면 목소리의 차이를 구별할 수 있다. 이런 정보를 분석하고 자동화하는 방법을 화자인식이라 부른다.

화자인식은 본인 인증의 한 수단으로 사용할 수 있다. 예를 들면 휴대폰 잠금 해제, TV, 에어컨 등의 기기 제어 등에 사용할 수 있다. 또한, 음성인식 과정에서 화자의 정보를 분석하면 개인화를 통해 콘텐츠 추천, 개인화 검색 등의 결과를 같이 줄 수 있으므로 서비스의 정확도를 좀 더 향상시키는 보조 수단으로 사용할 수 있다. 이 글에서는 화자 모델링에 사용하는 여러 방법들과 화자인식 평가 방법에 대해 알아본다.

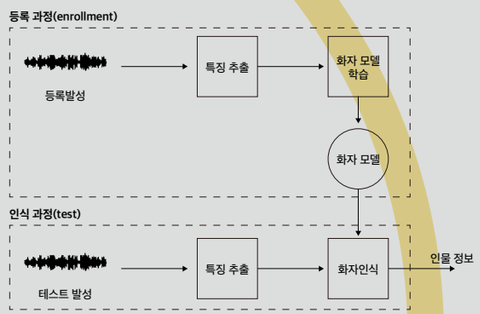
글 | 김명재 arth.mj@kakaocorp.com

SL2라는 음성 처리 회사에서 병역을 수행하고 있을 때, 음성 처리가 너무 어려워 정복해 보고 싶은 마음이 들었다. 하여 패기 있게 대학원에 진학하였으나 음성 처리는 여전히 어렵다. 어쩌다 보니 음성을 10년 넘게 다뤘지만, 아직도 배운 것보다 배워야 할 것이 많다. 항상 인공지능과 데이터 엔지니어링에 관심을 두고 있어 대가들의 움직임에 감탄하고 있다. 운 좋게 카카오에 입사하여 즐겁게 일하고 있는 개발자.

화자인식 개요

화자인식은 사람이 발성한 음성을 컴퓨터가 분석하여 음성의 인물 정보를 얻어 내는 과정을 말한다. 음성인식과 처리 과정은 유사하지만 음성인식은 발성한 음성에서 언어적 정보를 찾는 반면, 화자인식은 발성한 음성에서 인물 정보를 찾는다. 화자인식을 수행하기 위해서는 비밀번호 등록과 같이 발성한 음성을 통계적인 음향 모델(acoustic model)로 만드는 등록(enrollment)과정이 필요하며, 등록 과정에서 만들어진 음향 모델을 이용하여 인식(test) 과정에서 입력한 발성의 인물정보를 얻는다. [그림 1]은 화자인식 과정을 간단히 도식화한 그림이다.

[그림 1] 화자인식 개요도



화자인식은 얻은 정보를 분류하는 방법에 따라 화자 식별(speaker identification)[1]과 화자 확인(speaker verification)[2]으로 나눌 수 있다. 화자 식별은 등록된 사람들 중에서 발성한 사람을 찾는 과정이고, 화자 확인은 등록된 화자의 음성이 맞는지 결정하는 과정이다. 실제 화자인식기에서는 화자 식별과 화자 확인의 과정이 모두 필요하다.

화자 식별은 길이 T의 관측된 음성 특징 벡터열 $X=\{x_1,x_2,\cdots,x_T\}$가 주어졌을 때, 화자 집합 $S=\{\lambda_1,\lambda_2,\cdots,\lambda_k,\cdots,\lambda_S\}$에서 가능도(likelihood)가 가장 높은 모델 $\hat{S}$을 찾는 과정이며, 수식으로 표현하면 [수식 1]과 같다.

[수식 1]

$$\hat{S} = argmax_{1 \le k \le S} Pr(\lambda_k|X) = argmax_{1 \le k \le S} \frac{p(X \mid \lambda_k)Pr(\lambda_k)}{p(X)}$$

여기서 $Pr(\lambda_k|X)$는 음성 특징열이 주어졌을 때, 화자 모델 λ_k에 대한 가능도이며, 베이즈 룰(Bayes' rule)에 의해 우측 식으로 바꿀 수 있다. $p(X|\lambda_k)$은 화자 모델 λ_k에 대한 음성 특징의 가능도이다. $Pr(\lambda_k)$은 화자 λ_k가 등장할 사전 확률로 1/S로 모두 같다고 가정하고, $p(X)$는 모든 화자에게 동일하므로 [수식 2]와 같이 간략하게 표현할 수 있다.

[수식 2]

$$\hat{S} = argmax_{1 \le k \le S} Pr(\lambda_k|X)$$

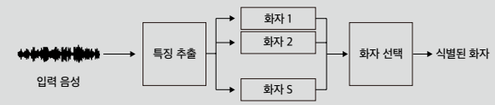
여기에 로그 함수를 사용하고 음성 특징 벡터열이 서로 독립적으로 관찰되었다고 가정하면 [수식 3]과 같이 표현할 수 있다.

[수식 3]

$$\hat{S} = argmax_{1 \le k \le S} \sum_{t=1}^{T} logP(X_t|\lambda_k)$$

[그림 2]는 화자 식별 과정을 나타낸 그림이다.

[그림 2] 화자 식별 개요도



화자 확인은 길이 T의 음성 특징 벡터열 $X=\{x_1,x_2,\cdots,x_T\}$가 관측되었을 때, 두 가지 가설 H_0와 H_1 중 한 가지를 선택하는 문제이다.

· H_0: X는 등록된 화자가 발성한 음성이다.

· H_1: X는 등록된 화자가 발성한 음성이 아니다.

이 문제를 수행하기 위해 화자 확인은 등록되지 않은 화자로 이루어진 배경 화자 모델을 필요로 한다. 화자 확인의 일반적인 접근 방법은 관측된 음성 특징 벡터열 X의 가능도 비율(likelihood ratio)을 사용하며 [수식 4]와 같이 표현할 수 있다.

[수식 4]

$$\frac{Pr(\lambda_k \mid X)}{Pr(\lambda_{UBM} \mid X)}$$

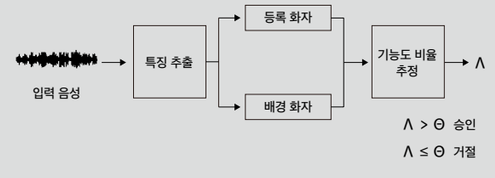
여기서 $Pr(\lambda_k|X)$는 음성 백터열 X가 주어졌을 때, 등록된 화자 모델 λ_k가 나올 확률이고, $Pr(\lambda_{UBM}|X)$는 음성 백터열 X가 주어졌을 때, 배경 화자 모델 (universal background model, UBM)λ_{UBM}가 나올 확률이다. [수식 4]에 베이즈 룰과 로그 함수를 적용하여 식을 다시 정리하면 다음과 같다.

[수식 5]

$$\Lambda(X) = log(X \mid \lambda_k) - log(X \mid \lambda_{UBM})$$

로그 유사도 비율 $\Lambda(X)$를 추정하여 기준점(threshold) θ보다 크면 등록된 화자 λ_k라 승인하고, 기준점 보다 작거나 같으면 등록된 화자가 아니라고 거절한다. [그림 3]은 화자 확인 과정을 보여준다.

[그림 3] 화자 확인 개요도

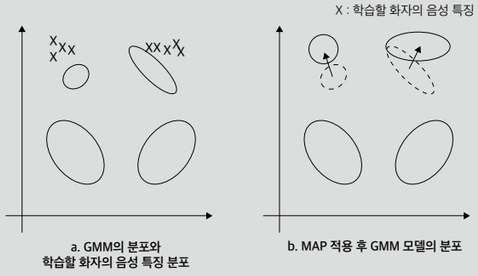


또한, 화자인식은 등록 발성 내용과 인식 발성 내용을 동일하게 제한하는 문장 종속 화자인식 방법과 인식 발성에 제한을 두지 않는 문장 독립 화자인식이 있다. 이 글에서는 화자인식에 널리 쓰이는 방법들과 평가 방법에 대해 알아본다.

화자 모델링

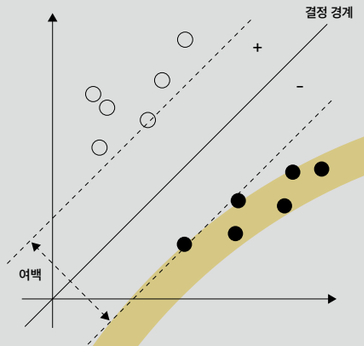
화자인식은 기본적으로 음성을 가우시안 혼합 모델(Gaussian Mixture Model, GMM)로 모델링 한다. GMM 방법은 개개인의 모델을 정교하게 만들기 위해 매우 많은 양의 음성 데이터를 필요로 한다. 그러나 수집할 수 있는 개개인의 음성 양은 제한되어 있기 때문에, GMM 방법으로는 높은 화자인식 성능을 기대하기 어려웠다. 이후, 다양한 사람으로부터 얻은 대량의 데이터로 GMM 모델을 학습하여 정교한 모델을 만들고, 개개인으로부터 얻은 소량의 등록 데이터를 최대 사후(Maximum a Posterior, MAP) 적응(adaptation) 방법을 통해 새로운 화자 모델을 만드는 GMM-UBM(Universal Background Model)[*3] 방법이 제안되었다. [그림 4]는 2차원의 GMM에서 소량의 화자 음성 특징으로 MAP 적응을 했을 때, 모델 분포의 변화를 보여 준다.

[그림 4] GMM의 분포와 MAP 적응 후의 분포



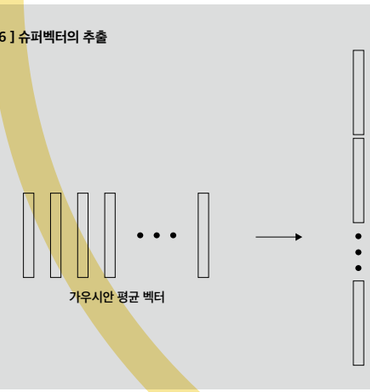
나이브 베이즈 (Naive Bayes) 기반의 분류 방법을 사용하는 GMM-UBM 방법에 SVM(Support Vector Machine)을 적용한 GMM-SVM방법이 제안되었다. [그림 5]는 2차원 공간에서 SVM의 이진 분류를 보여 준다. 실선은 부류를 결정하는 경계(hyperplane)이고, 점선에 위치하는 벡터가 서포트 벡터(support vector)이다. 서포트 벡터는 SVM의 결정 경계를 찾는 기준점이 되며, SVM은 서포트 벡터 간의 여백을 최대화하는 경계를 찾는 것을 목표로 한다.

[그림 5] SVM의 분류 예제



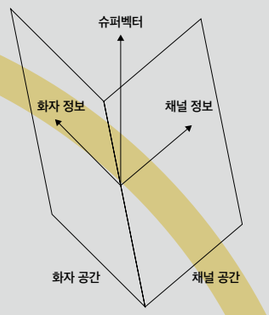
GMM-SVM의 입력으로는 슈퍼벡터(supervector)가 사용된다. 슈퍼벡터는 화자 적응된 GMM 모델의 평균 벡터를 연결하여 생성하는 하나의 매우 높은 차원의 특징 벡터이다. 슈퍼벡터는 매우 고차원의 특징이기 때문에 다루기 쉽지 않은 데다 매우 많은 메모리를 필요로 하며, 화자 정보 외에 다른 부가적인 채널 및 잡음 정보도 같이 표현된다. [그림 6]은 슈퍼벡터를 만드는 과정을 보여 준다. 예로 음성 특징이 60차원이고, GMM은 2,048개의 혼합 성분을 갖는다고 가정하면, 60 × 2,048 차원(12만 2,880차원)의 슈퍼벡터가 된다.

[그림 6] 슈퍼벡터의 추출



이와는 다른 방향으로 결합요인 분석(Joint Factor Analysis, JFA)[*4] 방법을 적용하여 슈퍼벡터에서 화자 정보(speaker factor)와 채널 정보(channel factor)를 분리하는 연구가 제안되었다. JFA 방법은 고윳값 분해(eigenvalue decomposition)에 기반하는데, 고유 분해 방법은 특정 데이터 집합으로부터 서로 수직(orthogonal)인 고유벡터(eigen vector)를 찾고, 이 고유벡터를 기저(basis)로 하는 변환 행렬을 통해 고유공간(eigenspace)으로 사영(projection)하는 방식으로 화자 정보를 얻는다. JFA에서는 변환 행렬의 정교한 추정을 위해, 많은 화자가 다양한 채널에서 녹음한 음성 데이터를 필요로 한다. 그러나 JFA는 한 화자가 다양한 환경에서 녹음한 데이터를 요구하기 때문에, 다량의 학습 코퍼스를 구축하기 어려운 문제점이 있다. [그림 7]은 JFA의 분해방법을 보여준다.

[그림 7] JFA의 벡터 분해



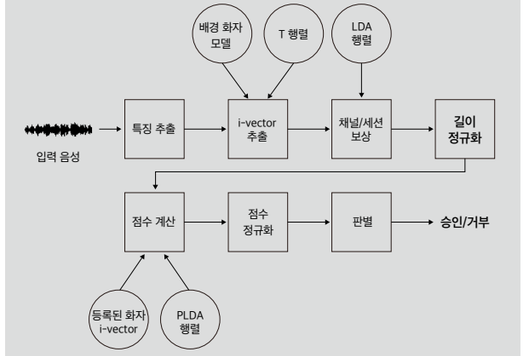
이런 방법을 개선하기 위해, 채널 특징을 제거하고, 변환 행렬(T-matrix)을 발성 단위로 처리하는 i-벡터(i-vector)[*5] 방법이 제안되었다. i-벡터 방법은 현재 화자인식 분야에서 가장 효율적인 특징 추출 방법으로 자리 잡았으며, 크게 추출(extraction), 채널/세션 보상(channel/session compensation), 길이 정규화(length normalization), 점수 계산(scoring), 점수 정규화(score normalization)로 나눌 수 있다. i-벡터 추출 방법은 앞서 소개한 JFA의 화자 정보 추출 방법과 동일하지만, 다른 점이 있다. JFA 방식은 화자 단위로 음성 데이터를 처리하는 반면, i-벡터 방식은 발성 단위로 음성 데이터를 처리한다는 점이다. 최근 i-벡터 추출과정 중, GMM posterior계산 과정을 DNN(Deep Neural Network)으로 대체한 방법도 제안되었다[*6].

채널/세션 보상은 시간이 지남에 따라 화자의 발화 상태가 조금씩 달라지는 현상과 서로 다른 마이크를 사용하여 음성 데이터를 받음으로써 생기는 왜곡, 배경 잡음 등을 감소시키기 위해 적용한다. 대표적인 채널/세션 보상 방법으로는 LDA(Linear Discriminant Analysis), NAP(Nuisance Attribute Projection) & WCCN(Within Class Covariance Normalization) 등이 있다.

i-벡터의 길이 정규화 방법은 점수 계산 방법과 큰 연관이 있다. i-벡터의 대표적인 점수 계산 방법으로 코사인 유사도(cosine similarity)와 PLDA(Probabilistic Linear Discriminant Analysis)[*7] 점수가 있다. 코사인 유사도 방법은 등록된 i-벡터와 인식하는 i-벡터의 각도를 점수화한 방법이고, PLDA 점수는 학습 발성과 인식 발성의 화자가 동일한 가정에서의 가능도와 다른 화자라 가정하는 상황에서의 가능도의 결과를 비율로 수치화 한다. PLDA의 가능도는 사전 분포(prior distribution)의 정의에 따라 HT-PLDA(Heavy-tailed PLDA)[*8]와 G-PLDA(Gaussian PLDA)[*9]로 나눌 수 있는데, HT-PLDA는 사전 분포를 Student's t 분포를 사용하며, G-PLDA에서는 Gaussian 분포를 사용한다. 코사인 유사도와 HT-PLDA는 i-벡터의 정규화를 수행하지 않으나, G-PLDA에서는 i-벡터에 길이 정규화를 수행한다. i-벡터에 길이 정규화를 수행하면 i-벡터의 분포가 가우시안 분포를 따르기 때문이다. i-벡터의 길이 정규화 방법에는 단순히 길이를 1로 만들어 주는 방법과, i-벡터의 요소(element)들의 크기를 기준으로 가우시안 정규화를 수행하는 순위 정규화(rank normalization), i-벡터를 단위 구체(unit sphere) 위에 위치하도록 변화 시키는 구면 길이 정규화(spherical length normalization)등이 있다[*10].

화자의 승인, 거부를 결정하는 기준점을 좀 더 명확히 찾기 위해 점수 정규화를 수행할 수 있다. 대표적인 점수 정규화 방법에는 Z-norm(Zero normalization), T-norm(Test normalization), ZT-norm(Z-norm + T-norm), S-norm(Symmetric normalization) 등이 있다. [그림 8]은 i-벡터/PLDA 기반의 화자 확인 과정을 보여 준다.

[그림 8] i-vector/PLDA 기반의 화자 확인 과정

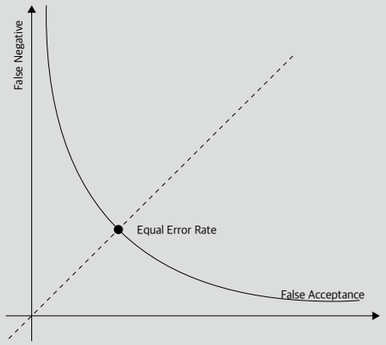


평가 방법

화자 확인의 평가 방법에는 EER(Equal Error Rate)과 minDCF(minimum Detection Cost Function)가 있다. 화자 확인은 [그림 9]에서 보듯이 특정 기준점(threshold)에 의해 오인식(false acceptance, FA)과 부정오류(false negative, FN)의 발생량이 달라지기 때문이다. 여기서 FA는 등록되지 않은 화자가 발성했지만 등록된 화자로 잘못 승인하는 경우를 말하며, FN은 등록된 화자가 발성했지만 등록된 화자가 아닌 것으로 잘못 거절한 경우를 말한다. 기준점을 높게 잡으면 FA가 적게 발생하지만 반대로 FN이 높아진다. 반대로 기준점을 낮게 잡으면 FA가 많이 발생하는 반면 FN이 적게 발생한다. 이러한 이유로 화자 확인은 EER과 minDCF

같은 평가 방법을 이용한다. EER은 FA와 FN가 동일하게 발생하는 기준값에서의 오류율을 말하며, [그림 9]는 EER이 정해지는 위치를 보여 준다.

[그림 9] Equal Error Rate



minDCF는 미국 국립표준기술연구소(National Institute of Standards Technology, NIST)*11에서 주관하는 화자인식 대회에서 사용하는 평가 방법이며, minDCF는 [수식 6]과 같다.

[수식 6]

$$C_{Det} = C_{Miss} \times P_{Miss|Target} \times P_{Target} + C_{FalseAlarm} \times P_{FalseAlarm|NonTarget} \times (1 - P_{target})$$

오류 검출(false alarm)은 오인식(FA)와 같고, Miss는 오류 검출(FN)과 같다. [수식 6]에 사용되는 파라미터는 [표 1]과 같다.

[표 1] Detection Cost Model Parameters*12

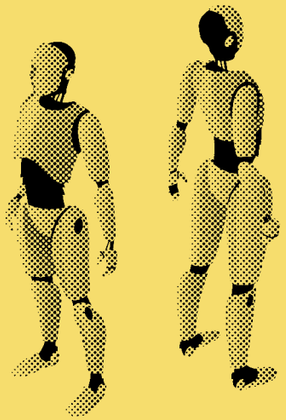
C_{Miss}	$C_{FalseAlarm}$	P_{Target}
10	1	0.01

마치며

지금까지 고전적인 화자인식 방법과 최근에 사용하는 화자인식 방법들을 살펴보았다. 화자인식은 기존의 문제들을 조금씩 해결하는 방향으로 발전하고 있다. 최근에 화제가 된 딥러닝을 부분적으로 적용하는 시도들뿐만 아니라 end to end 방식으로 성능을 향상하려는 시도들도 많이 소개되고 있다. 그러나 화자인식의 특성상 한 화자에 대해 많은 정보를 얻기 힘들기 때문에, i-벡터 / PLDA 방법이 여전히 최고의 자리를 유지하고 있다. 현재 화자인식은 기존의 통계적인 방법과 딥러닝을 적절히 조합하는 방법으로 진화하고 있으며, 음성을 사용하는 개인화된 서비스와 융화되기를 기다리고 있다.

***1** 논문 | Reynolds, D. A. & Richard C. R. (1995). Robust text-independent speaker identification using Gaussian mixture speaker models, IEEE transactions on speech and audio processing. ***2** 논문 | Reynolds, D. A. (1995). Speaker identification and verification using Gaussian mixture speaker models, Speech Communication. ***3** 논문 | Campbell, W. M., Strurim, D. E., Reynolds, D. A. & Solomonoff, A. (2006). SVM based speaker verification using a GMM supervector kernel and NAP variability compensation, ICASSP. ***4** 논문 | Kenny, P., Boulianne, G., Ouellet, P. & Dumouchel, P. (2005). Factor analysis simplified [speaker verification applications], ICASSP. ***5** 논문 | Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P & Ouellet P. (2011). Front-end factor analysis for speaker verification, IEEE Transactions on Audio, Speech, and Language Processing. ***6** 논문 | Snyder, D., Garcia-Romero, D. & Povey, D. (2015). Time delay deep neural network-based universal background models for speaker recognition, ASRU. ***7** 논문 | Prince S. J. & Elder J. H. (2007). Probabilistic linear discriminant analysis for inferences about identity, ICCV. ***8** 논문 | Kenny, P. (2010). Bayesian Speaker Verification with Heavy-Tailed Priors, Odyssey. ***9** 논문 | Garcia-Romero, D. & Espy-Wilson, C. (2011). Analysis of i-vector Length Normalization in Speaker Recognition Systems, Interspeech. ***10** 논문 | Shum, S., Dehak, N., Dehak, R. & Glass, J. (2010). Unsupervised Speaker Adaptation based on the Cosine Similarity for Text-Independent Speaker Verification, Odyssey. ***11** 참고 | NIST, "The NIST Year 2010 Speaker Recognition Evaluation Plan," Odyssey, 2010. ***12 참고 |** [수식 6]에 사용하는 파라미터 [표 1]은 NIST SRE(Speaker Recognition Evaluation) 평가 년도마다 조금씩 차이가 나며 [표 1]은 NIST SRE 10을 기준으로 한다.

AI 현장의 이야기



카카오는 국내 인터넷 업체 중 처음으로 2015년 6월 뉴스 서비스에 인공지능 체계를 전면 적용했습니다. 카카오의 도전은 성공적이라는 평가를 받고 있습니다. 카카오 인공지능 플랫폼인 카카오I의 추천 엔진은 또 한 번의 진화를 했습니다. 카카오I 추천 엔진의 개선 배경 및 양상을 설명한 성인재 님의 글은 미디어와 IT의 이슈를 꾸준히 정리하고 계신 분들에게 많은 관심을 받을 것으로 예상됩니다. 국내 AI업계에서 저명한 신정규 님은 '딥러닝과 데이터'라는 제목의 글로 또 다른 현장의 이야기를 들려 주셨습니다. 카카오브레인의 이수경 님은 현재 AI에서 가장 핫한 이슈인 알파고 제로를 기존 알파고와 비교해서 설명해 주셨습니다.

카카오I 추천 엔진의 진화: 뉴스 적용 사례를 중심으로

2015년 6월, 카카오는 인공지능에 의한 뉴스 서비스 체계를 모바일 다음에 전면 도입했다. 국내 인터넷 업계에서는 최초의 일이었다. 그리고, 올해 4월 카카오는 PC(Personal Computer)에도 같은 인공지능 시스템을 도입했다. 이러한 확대 적용은 지난 2년 간 확인된 인공지능 체계의 성과가 반영된 결과다. 그리고 2017년 9월, 카카오는 뉴스 서비스에서 또 다른 혁신을 시도한다. 이용자가 실제 콘텐츠를 읽은 시간을 파악하는 지표를 기존 알고리듬에 결합시킨 것이다. 새로운 변화의 등장 배경, 그리고 결합 방식에 대한 상세한 설명을 이번 글에 담았다.

글 | 성인재 tevin.sung@kakaocorp.com

카카오I의 추천 엔진 프로젝트인 루빅스를 리딩하고 있습니다.
SAP Labs Korea와 네이버를 거쳐 현재 카카오에 재직하면서, 컨텐츠 추천, 정답 추천, 문맥광고, 연관검색어 등의 프로젝트를 리딩하고 AI 및 머신러닝 알고리듬을 설계하였습니다.

카카오I의 추천 엔진 : 뉴스 적용 사례(루빅스)

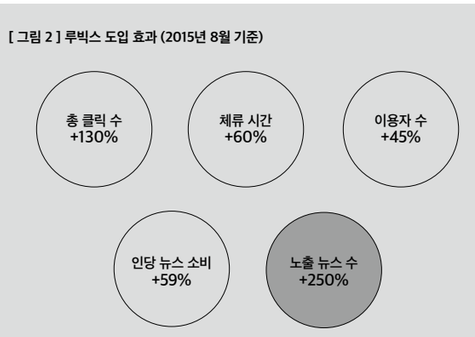
루빅스 시스템의 개요

루빅스(Real-time User Behavior-based Interactive Content recommender System, RUBICS)는 2015년 6월부터 다음모바일메인 뉴스에의 적용을 시작으로, 현재는 다음모바일메인의 뉴스/연예/스포츠를 비롯한 대부분의 탭, 다음PC메인의 뉴스/연예/스포츠, 카카오톡 채널 등에 적용되어 서비스되고 있는, AI 기반의 콘텐츠 추천 시스템이다. 루빅스의 도입으로 이용자들은 동일한 뉴스 서비스 화면에 노출되는 것이 아니라 각 이용자의 관심사에 맞는 뉴스들로 이루어진 개인화된 뉴스 서비스 화면을 소비할 수 있게 됐다.

[그림 1] 루빅스 개요

루빅스는 2015년 6월부터 다음 모바일 메인 뉴스에 처음 적용됐다. 루빅스 도입 이후, 다음 뉴스의 이용량이 증가되었을 뿐만 아니라, 제공되는 뉴스의 다양성도 확대되었다. 루빅스를 통해, 뉴스 콘텐츠의 다양화, 뉴스 이용자 규모, 뉴스 체류 시간이 모두 증가하는 선순환 구조가 형성됐다고 볼 수 있다. 이에 카카오는 루빅스를 모바일 다음 메인의 뉴스/연예/스포츠를 비롯한 대부분의 탭, 다음 PC 메인의 뉴스/연예/스포츠, 카카오톡 채널 등까지 적용시켰다.

[그림 2] 루빅스 도입 효과 (2015년 8월 기준)



루빅스는 실제 콘텐츠 서비스에서 나타난 특성을 면밀히 분석한 결과를 기반으로, 다양한 추천 알고리듬을 앙상블(ensemble)하여 랭킹(ranking)된 추천 콘텐츠를 제공한다. 루빅스에 적용되어 있는 알고리듬은 크게 사용자 그룹 맞춤 추천과 개인화 추천의 두 부류로 구분될 수 있다. [표 1]

[표 1] 루빅스의 콘텐츠 추천 방식

	사용자 그룹 맞춤 추천	개인화 추천
특성	· 콘텐츠들의 선호도를 실시간으로 측정하여, 다수의 사용자가 관심을 보인 기사 추천 · Cold Start 사용자에게 기사 추천 가능 · 상대적으로 기사의 다양성이 증대되어, 메인판 구성에 적합	· 사용자 개개인이 관심 있을 콘텐츠를 추천 · 기사를 본 기록이 있는 사용자에게만 기사 추천 가능
알고리듬	멀티암드밴딧(Multi-Armed Bandit)을 이용한 사용자 그룹별 추천	협업필터링(Collaborative Filtering), 주제어 기반(Topic Keyword-based) 등
적용서비스	다음 모바일 메인의 뉴스를 비롯한 대부분 탭, 다음PC메인 뉴스/연예/스포츠, 카카오톡채널	카카오톡채널

루빅스의 추천 알고리듬에 대해서는 이미 문서[1]로 공개된 바 있으므로, 여기서는 이 정도로 설명을 마치도록 한다. 다음 장에서는 루빅스에 최근 새롭게 도입된 지표 및 이를 이용한 알고리즘에 대해 주로 다루고자 한다.

DRI(Deep Reading Index)에 대한 설명

도입 배경

루빅스 적용 후 2년 넘게 사용자 선호도의 메인 지표는 CTR(click through rate)였다. CTR과 같은 클릭 지표는 루빅스뿐만 아니라, 일반적으로 널리 알려진 콘텐츠 추천 알고리듬들의 메인 지표로 사용되고 있다.

하지만 클릭 지표는 컨텐츠의 "제목"에 대한 사용자의 반응만을 반영하고 있고, 소비 선택 빈도 외에 콘텐츠가 어떻게 소비되는지는 반영하지 못한다. 따라서 클릭 지표만을 이용하여 콘텐츠를 추천하는 경우, 콘텐츠의 소비 양상과 태도를 온전히 반영하기는 어렵다.

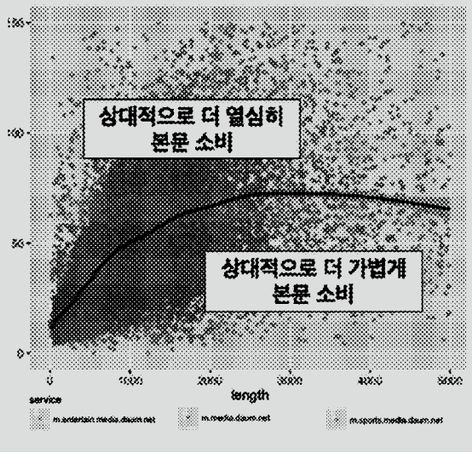
실제 콘텐츠가 어떻게 소비되었는지에 대한, 즉 "본문"에 대한 사용자의 반응에 대한 측정은 지표 그 자체로서도, 추천 알고리즘에 이용하기 위해서도 절실히 필요하였다. 다양한 고민 끝에 결국 "사용자들이 본문을 열심히 읽은 정도"를 가늠할 수 있는 지표인 DRI(deep reading index, 열독률 지수)를 고안해냈다.

필요성 및 정의

이용자의 체류 시간은 같아도 취득하는 정보량이 다를 수 있기 때문에, 콘텐츠가 얼마나 열심히 읽혔는지를 체류 시간으로 바로 가늠하기는 어렵다. 예를 들어 본문 500자, 이미지 1개인 콘텐츠와 본문 1,000자, 이미지 5개인 콘텐츠의 평균 체류 시간이 모두 50초일 때, 이용자들이 두 콘텐츠를 동일한 수준으로 열심히 봤다고 하긴 어렵다. 모바일 다음의 뉴스/연예 스포츠 기사의 본문 길이에

따른 체류 시간 중앙값을 플로팅(plotting)한 [그림 3]을 살펴 보면, 콘텐츠의 정보량을 나타내는 값의 하나인 본문 길이가 체류 시간과 상관 관계에 있음을 알 수 있다. 이러한 경향성에 부합할 수 있도록 콘텐츠에 포함된 정보량에 따른 기대 체류 시간과 특정 콘텐츠 체류 시간을 동시에 고려할 수 있는 새로운 지표로 개발된 것이 DRI이다. DRI는 기대 체류 시간 대비 해당 콘텐츠의 체류 시간의 상대적인 크기로 정의된다. 다시 말해, "상대적인 체류 시간"을 통해 사용자의 본문 선호도를 측정한다.

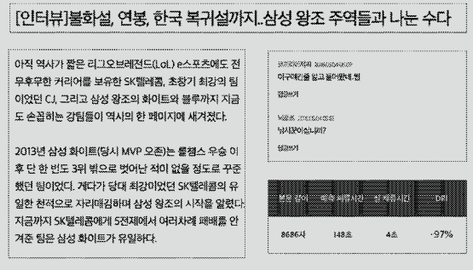
[그림 3] 콘텐츠 본문 길이에 따른 체류 시간



DRI 적용 효과의 예시

DRI는 콘텐츠가 본문에서 제목에서 예상되던 바를 충족시키지 못한 경우를 발견해 낼 수 있다. [그림 4]는 사용자들이 제목과 본문의 불일치를 체감한 경우가 많은 기사의 예이다. 야구 관련 기사로 예상하고 클릭을 했으나, 실제 본문 내용은 e스포츠라서 이탈하는 상황이 많이 발생하는 것으로 보이는 뉴스 기사이다. 이 기사의 DRI는 -97%로 매우 낮게 나타나는 것을 알 수 있다.

[그림 4] 콘텐츠 특성에 따른 DRI 예



DRI-CTR 앙상블 기반 루빅스 추천

DRI-CTR 앙상블 소개 및 랭킹 방식 비교

기존 루빅스 추천의 기반이었던 CTR에 DRI가 추가된 루빅스 알고리듬을 소개하고자 한다. DRI를 CTR과 앙상블(ensemble)하여, DRI의 총합을 최대화하도록 목적함수(objective function)가 정의되었다. CTR 기반, DRI 기반, DRI-CTR 앙상블 기반으로 사용자 그룹 맞춤 추천을 한 경우는 각각의 특성은 [표 2]와 같이 정리된다.

[표 2] 랭킹 방식 비교

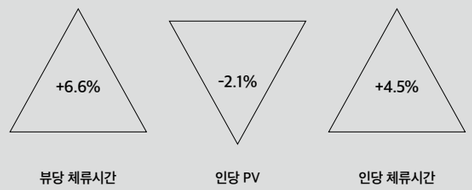
랭킹	CTR 기반	DRI 기반	DRI - CTR 앙상블(Ensemble) 기반
시간에 따른 변동성	높음	낮음	중간
반영되는 사용자 반응	제목	본문	제목과 본문

DRI-CTR 앙상블 기반은 제목과 본문에 대한 사용자 반응을 모두 반영할 수 있는 장점을 갖고 있다. DRI는 기대 체류 시간 대비 해당 콘텐츠 체류 시간의 상대적인 크기로 정의되고, DRI-CTR 앙상블 기반 추천 랭킹은 DRI의 합을 최대화하기 때문에, 결과적으로 체류 시간의 간접적인 증가를 기대할 수 있다.

DRI 도입의 효과

CTR랭킹 대비 DRI-CTR 앙상블 랭킹으로 사용자 그룹 맞춤 추천 시의 지표 변화를 알아보기 위해 실험을 진행한 결과, [그림5]와 같은 지표 변화를 보였다. 모바일 다음 메인의 뉴스 / 연예 / 스포츠 전체를 대상으로 측정된 지표이다.

[그림 5] CTR랭킹 대비 DRI-CTR 앙상블 랭킹의 지표 변화량



DRI가 랭킹 요소에 추가되면서, 끌리는 제목을 가진 기사들이 상대적으로 적게 추천되기 때문에 '1인당 평균 페이지뷰(page view)'가 소폭 하락했다. 그러나, 사용자들이 보다 열심히 읽을 만한 기사를 제공해 주기 때문에 "1뷰(view) 당 평균 체류 시간"이 상승하여, 결과적으로는 '1인당 평균 체류 시간'이 상승했다. DRI-CTR 앙상블 기반 추천 랭킹을 통해, 제목과 본문에 대한 사용자의 반응을 모두 기사 추천에 반영할 수 있게 되었으며, 더불어 총 체류 시간도 증가하게 되는 것이다.

향후 목표 및 방향

지금까지 루빅스의 도입 배경과 추천 알고리즘, 그리고 DRI란 무엇인지, 루빅스 추천에 DRI를 도입한 후의 효과 등을 살펴보았다. 현재도 사용자 반응 데이터를 기반으로 꾸준히 루빅스의 성능 평가가 진행되고 있으며, 그 결과에 따라 알고리즘 및 시스템 개선작업이 지속적으로 진행되고 있다.

2017년 9월부터, 기사 제목과 본문에 대한 사용자 반응을 모두 반영하여 추천하는 DRI-CTR앙상블 알고리즘이 루빅스에 도입되었고, 다음모바일메인 뉴스/연예/스포츠에 적용되어 서비스 중이다. 향후 루빅스에 DRI를 도입함으로써 나타난 체류시간 상승이 재방문률 상승으로 이어지는지에 대해 지속적으로 측정할 예정이다. 체류시간의 상승은 곧 사용자들의 서비스 만족도가 상승했다는 것을 의미할 수 있고, 만족도의 상승으로 인해 보다 더 자주 서비스를 방문하게 됨으로써, 결과적으로 재방문률의 상승이 나타날 수도 있다. 예상과 같은 결과가 나타난다면, 사용자에게 보다 높은 가치를 제공함으로써 재방문을 이끌어낸, 좋은 사례가 될 수 있을 것이다.

*1 참고 | 박승택, 성인재, 서상원, 황지수, 노지성, 김대원. (2017). 기계학습 기반의 뉴스 추천 서비스 구조와 그 효과에 대한 고찰: 카카오의 루빅스를 중심으로. 사이버커뮤니케이션학보, 34권 1호, 5-48.

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

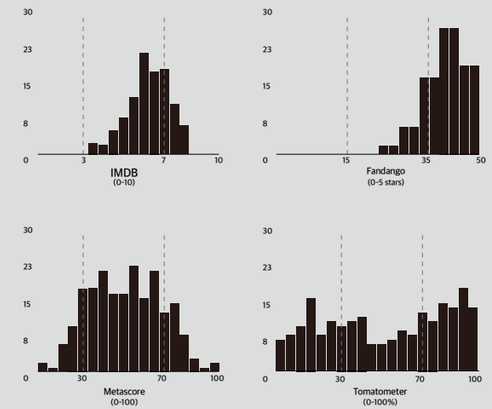
미디어 추천 시스템을 만드는 경우를 가정하자. 개인화 추천 시스템으로, 어떤 사용자가 어떤 콘텐츠를 얼마나 좋아할 것인지를 예측하는 모형을 만드는 것이 목표이다. 모형의 훈련 데이터로 가장 쉽게 사용할 수 있는 것은 랭킹 데이터이다. 수많은 사용자들이 영화 및 드라마에 점수를 매겨 놓은 랭킹 데이터를 가정해 보자. 네이버 영화 평점은 10점 만점 시스템, 왓챠의 시스템은 5점 만점 별표 시스템이다. (중간에 별표 반 개를 가능하게 하여 10점 시스템으로 바뀌었지만, 이러한 경우는 뒤에서 따로 다룰 것이므로 여기에서는 논외로 한다) 넷플릭스의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

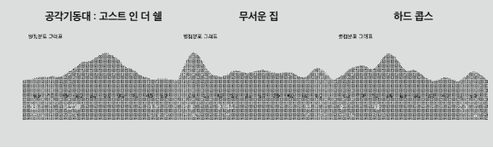
서로 다른 스케일의 데이터를 정규화하여 하나의 데이터 세트로 만들어 보자. 어느 데이터 세트를 기준으로 스케일을 맞출 것인가? 아티팩트(artifact)를 추가하지 않기 위해서는 더 낮은 해상도의 데이터 세트로 스케일하는 것이 일반적이다. 그렇다면 위에서 예로 든 데이터 세트의 경우 좋아요/아니요의 이진 데이터로 스케일해야 한다. 이 경우 10점 만점의 데이터는 몇 점을 기준으로 좋아요/아니요로 변환해야 할까? 만약 데이터가 바이모달 분포(bimodal distribution)[*7]를 따르고, 최고점이 두 개라면 나누기 쉬울 것이다.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

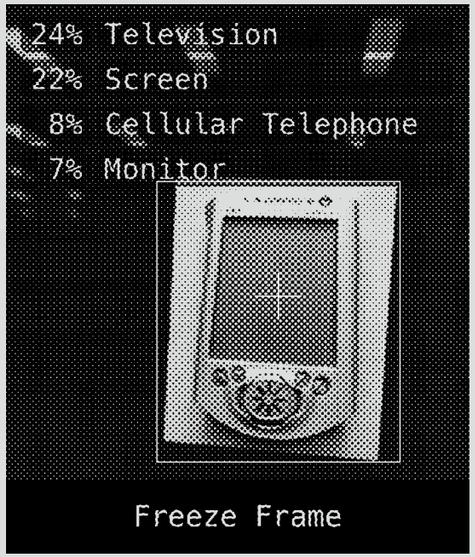
심층 신경망의 대두에는 심층 신경망을 훈련할 수 있는 충분한 (엄청난) 양의 데이터가 뒷받침되었다. 그런데 그 데이터들이 어디에서 왔을까? 사실 '어디'보다는 '언제'가 더 적합한 질문이다. 거의 모든 빅데이터는 최근에 생성되었다. 빅데이터는 '더 다양한' 데이터를 '생성'하고 '기록'하는 과정을 전산화하는 과정의 부산물이다. 그런데 빅데이터의 증가 추세는 지수적 증가에 가깝다. 비교적 오래되고 계량화된 주식 거래 데이터의 경우를 살펴보자. 뉴욕 증권 거래소의 1993년 거래 틱 데이터의 총 용량은 4.25기가이다. 1998년에는 20기가가 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

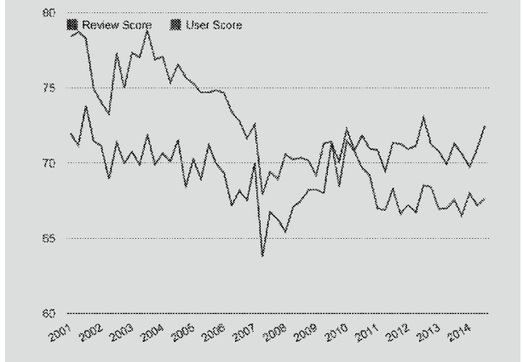
인공 투자자는 주식 투자자들의 꿈이다. 기계 학습은 알고리듬 매매에 오래전부터 사용되어 왔다. 인공 신경망의 투자 응용도 비교적 오래전부터 적용된 분야이다. 기계 학습이 패턴 인식에 강한 특성이 있기 때문이다. 그런데 지속적으로 엄청난 돈을 벌어들인 단일 모형은 등장하지 않았다[*29].

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 값들만 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

알파고 제로 vs 다른 알파고

세간에 알려진 알파고는 총 4가지 버전으로 존재한다. 지난 2015년 10월 천재 바둑 기사 판 후이(Fan Hui) 2단을 이기고 2016년 네이처(Nature)에 실린 버전인 알파고 '판(Fan)', 2016년 3월 이세돌 9단을 4대 1로 이긴 알파고 '리(Lee)', 커제 9단과 대결에서 3:0 완승을 거둔 알파고 '마스터(Master)', 그리고 2017년 네이처를 통해 공개된 알파고 '제로(Zero)'가 바로 그것이다. 참고로 알파고 리, 마스터, 제로의 구조와 학습법은 이번 논문에서 새롭게 소개됐다.

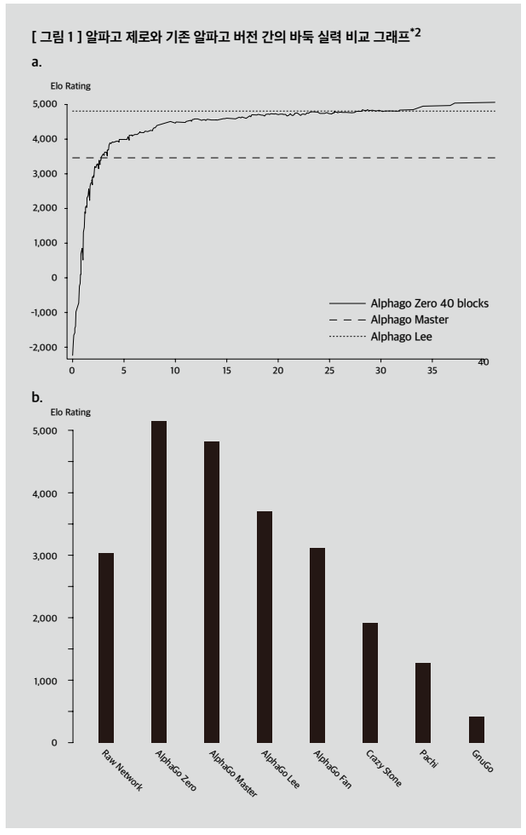
글 | 이수경 samantha.lee@kakaobrain.com

2016년 3월 알파고와 이세돌 9단이 펼치는 세기의 대결을 두눈으로 목도한 이후 인공지능을 제대로 공부해봐야겠다고 결심했다. 인공지능 본진이자 연구소인 카카오브레인으로 걸어들어온 이유다. 인공지능 기술과 이로 인해 바뀔 미래 사회를 다루는 글을 통해 사람들과 소통하고 싶다.

기술감수 | 천영재 yeongjae.cheon@kakaobrain.com

알파고 제로는 이전(前) 세대와 비교했을 때 월등한 성능을 자랑한다. 순위 산출에 사용되는 엘로(Elo) 점수*1를 기준으로 했을 때 알파고 제로는 5,185점을 보유하고 있다[그림 1]. 알파고 마스터(4,858점)는 327점, 알파고 리(3,739점)와는 1,446점, 알파고 판(3,144점)과는 2,041점의 격차가 있었다. 엘로 점수에서 800점 이상 차이 나면 승률이 100%라는 것을 고려했을 때, 알파고 제로가 현존하는 인공지능 바둑 컴퓨터로서 최정상급이라는 점을 부인하긴 어렵다.

[그림 1] 알파고 제로와 기존 알파고 버전 간의 바둑 실력 비교 그래프*2



이번 알파고 제로 논문이 시사하는 바에 대해 들어보고자 카카오브레인의 천영재 연구원과 감동근 아주대학교 교수로부터 자문을 구했다. 두 사람은 알파고 제로 이전과 알파고 제로 간 3가지 차이가 있다고 말했다.

첫 번째 : 신경망 통합

알파고 제로 전(前)세대들은 정책망(policy network)과 가치망(value network) 이라는 2가지 종류의 신경망을 갖췄다. 이 두 신경망을 구축한 이유는 앞으로 진행될 경기를 미리 여러 번 진행해보고, 승리할 가능성이 높은 수만을 효과적으로 탐색하기 위해서다.

실제 바둑 한 경기당 2×10^{170}이 넘는 경우의 수가 존재하는데, 이는 전세계에서 가장 큰 규모의 슈퍼컴퓨터로도 다 계산하기 어려운 규모다. 따라서 시뮬레이션 횟수를 줄이면서도(깊이), 승률이 높은 수(너비)를 찾는 탐색 알고리듬 구축이 관건이라고 볼 수 있다.

정책망은 바둑판 상태를 분석하여 361(=19×19)가지 경우의 수중에 가장 수읽기 해볼 만한 몇 가지 수를 선택한다. 가치망은 어떤 수를 두었을 때 그 후에 일어날 미래 대국을 시뮬레이션해본 뒤 그 결과로부터 승패를 예측한다. 보다 쉽게 이야기하자면 정책망은 '다음에 둘 수'를, 신경망은 '판세(승패)'를 예측한다.

이번 알파고 제로에서는 이 정책망과 가치망을 하나의 네트워크로 구현했다. 이 구조는 두 가지 의미를 내포한다. 하나는 자신만의 바둑 이론을 하나의 신경망으로 표현했다는 것이고, 또 하나는 성능을 높이는 방식을 선택했다는 것이다. 예측 정확도는 다소 낮아지나 값 오류(value error)는 낮추고 플레이 성능은 높일 수 있게 된다. 천영재 연구원은 "딥러닝 초기에 제안된 단순한 CNN 구조에서, 비교적 최근 제안된 레스넷(ResNet)*3으로 네트워크 구조를 변경해 성능 개선을 얻었다"며 "아울러 하나의 네트워크에서 정책망과 가치망을 한 번에 테스트함으로써 같은 시간 내 2배 더 많은 추론(inference)이 가능해졌고, 궁극적으로 트리 탐색에서 이득을 보았다"고 분석했다. 앞선 구조 변경은 엘로 점수를 대략 600점 올릴 수 있었던 원동력 중 하나로 간주된다.

두 번째 : 무(無)에서 유(有)로의 학습

전(前) 버전의 알파고에선 15만 건의 기보(棋譜, 한판의 바둑을 두어 나간 기록)로부터 3,000만개의 수를 입력받아 지도학습(supervised leaning) 방식으로 정책망을 학습해 나갔다. 이렇게 다음 수를 예측하는 정확도를 57%까지 끌어올린 이후, 알파고는 강화학습(reinforcement learning)을 통해서 정책망과 가치망을 다듬어 나갔다. 이 단계에선 스스로 새로운 전략을 발견하고, 바둑에서 이기는 법을 학습했다.

반면, 알파고 제로는 인간이 만든 기보나 수를 전혀 학습에 사용하지 않았다. 오로지 바둑 규칙만을 가지고 자가 대국을 두며 처음부터 끝까지 인간의 도움 없이, 스스로 바둑 이치를 터득해나갔다.

인간으로부터 전혀 배운 것이 없는 알파고 제로는 인간의 선입견과 한계로부터 자유를 얻었다. 그 덕분에 자신만의 독특한 정석(공격과 수비에 최선이라고 인정되는 수를 두는, 일련의 순서)을 개발했다. 사람이라면 바둑 세계에 입문하자마자 배우는 '축'의 개념을, 알파고 제로는 정작 학습이 상당히 진행된 다음에 발견하기도 했다.

감동근 교수는 "강화학습만으로 개발한 알파고 제로는 인간과는 전혀 다른 바둑을 둘지도 모른다고 생각했으나 오히려 인간이 지난 2,500년간 찾아낸 바둑의 수법이 아주 허황한 것이 아님을 보여줬다"고 평가했다.

다만 실전에서 인간 프로기사를 이길 수 있을지에 대해서는 의견이 분분하다. 알파고 제로는 가장 간단한 바둑 규칙(Tromp-Taylor rule)으로 개발됐다. 대표적으로 실전에서는 허용된 동형반복을 학습하지 못했다. 실전에서 삼패를 만들게 된다면 인간이 알파고 제로를 가지고 놀 수 있다는 것을 함의한다. 감 교수는 "이 때문에 구글 커제와 대결이 있었던 올해 5월까지도 구글 딥마인드팀이 알파고 제로에 대해 확신을 갖지 못한 것 같다"고 추측했다.

강화학습이라고 설명하는 부분은 다소 주의 깊게 볼 필요가 있다. 알파고 제로는 자가 대국한 결과를 가지고 네트워크를 지도학습을 반복, 최종적으로 높은 성능의 네트워크를 학습한다. 이는 일반적으로 보상(reward)만을 가지고 네트워크를 학습시키는 강화학습과는 다소 차이가 있다. 강화학습이 지도학습과 대비되는 가장 큰 특징은 학습 데이터가 주어지지 않는다는 점이다.

세 번째 : 효율적인 학습과 테스트

알파고 판과 알파고 리는 각각 1,202개의 CPU와 176개의 GPU를, 1,202개의 CPU와 48개의 TPU를 분산처리해 하나의 컴퓨터처럼 묶은 뒤 대국을 진행했다. 반면, 알파고 마스터와 알파고 제로는 4개의 TPU만을 가진 컴퓨터(싱글 머신)로 경기에 임했다.

이는 알파고 마스터와 제로가 기존 인공지능 바둑에서 당연하게 받아들였던 많은 부분을 제거, 속도 개선 효과를 얻었기에 가능했다. 대표적으로, 네트워크의 입력으로, 활로(liberty)의 수, 사석의 수, 불가능한 수의 위치 등 사람이 정의한 다양한 특징(hand-crafted features)은 사용하지 않고 단지 흰돌과 검은돌의 위치 정보만을 사용했고, 바둑을 끝까지 빠르게 두어보는 롤아웃(roll-out)을 제거했다. 또한, 네트워크의 고도화로 트리탐색의 효율을 높였다. 결과적으로 CPU 자원사용이 절대적으로 줄었고, 더 적은 수의 GPU(혹은 TPU)만으로도 이전 버전의 성능을 뛰어넘을 수 있었다.

다만 감 교수는 "TPU 몇 개를 갖춰서 학습시킨 지 불과 몇 시간 만에 이전 알파고 버전과 인간을 뛰어넘었다"며 접근 방식에 대해 우려를 표했다. 알파고 제로를 기준으로 대국에는 단일 머신(4 TPU)을 활용했지만, 학습에는 64 GPU와 19 CPU를 활용한 것으로 파악된다. 이는 하나의 실험 환경에서 이같은 컴퓨팅 자원을 활용했다는 의미로, 조작변인을 조금씩 바꿔가며 수십, 수백개의 실험을 병렬로 수행하려면 많은 양의 GPU(혹은 TPU) 자원이 더 필요할 수도 있다.

그저 단순히 원점(zero base)에서 학습을 시작한 지 수십 시간 만에 알파고 제로가 이전 버전을 뛰어넘은 것은 아니라는 의미다. 어마어마한 컴퓨팅 자원과 인력을 가지고도 최적의 인자(parameter)를 찾기 위해서는 최소 수개월이 필요할 수 있다.

추가로, 엘로 점수가 인간 프로 선수보다 1500점 정도 높다고 해서 5~6점 깔아야 한다는 시각은 근거가 약하다. 감 교수는 "아마 5단인 나는 호선(互先)*4으로 승률이 50%인 상대한테 2점을 접고도 10판을 둔다면 그 중 한판은 이기리라 기대할 수는 있어도, 세계 랭킹 1위인 커제(柯潔) 9단과 한국 1위 박정환 9단이 두면 2점이 아니라 정선(定先)*5으로도 10대 0이다"라고 설명했다.

알파고 제로의 공개로 가까운 미래에 인공지능을 탑재한 기계가 인간을 지배하는 것이 아니냐는 우려가 더 커졌다. 반면 이를 제대로만 활용한다면 인류가 당면한 각종 사회문제를 해결할 키가 될 것이라는 장밋빛 미래도 그려지고 있다. 분명한 건 알파고 제로가 19×19라는 작은 바둑판 내 문제를 푸는 최강자라는 점을 부인할 수 없다는 것이다. 다만 지구상에 존재하는 문제는 이보다 더 복잡한 경우의 수로 점철되어 있다는 점이다. 알파고 제로의 탄생에 환호하긴 아직 이르다. 아직 우리 인간이 가야 할 길은 멀고 풀어야 할 문제는 더 많다.

*1 참고 | 바둑 실력을 수치화 한 점수. 엘로 점수 차이가 200점 이상인 두 AI 맞붙는다면, 점수가 높은 AI가 이길 확률은 75%이다. 366점 차이라면 90%, 677점 차이는 99%, 800점 이상의 격차인 경우, 우위의 AI가 이길 확률은 사실상 100%가 된다. *2 논문 | Silver, D. et al. (2017). Mastering the Game of Go without Human Knowledge (p.13), doi:10.1038/nature24270. *3 참고 | CNN는 얀 레쿤 교수가 1989년 개발한 구조를 토대로 한다. 2012년 ILSVRC 이미지인식 대회에서 힌튼 교수팀의 알렉스넷(AlexNet)이 놀라운 성능 개선을 보이며 CNN에서 폭발적인 연구 성장이 이어져 왔다. 이후 딥러닝이 복잡한 문제를 해결하는 열쇠라는 게 밝혀지면서 이후로도 딥러닝 연구가 이어져오고 있다. VGGNet, 구글넷(GoogLeNet), 레스넷 등이 2011년 26% 수준의 인식오차율을 3.6%까지 낮춘 CNN 개량판이다. 그 중 레스넷은 마이크로소프트가 개발한 것으로, 이미지 인식 네트워크 중에서도 인기가 많다. *4 참고 | 호선은 바둑 플레이어 간 실력이 막상막하일 경우, 돌가리기를 통해 흑백을 정한 다음 시작하는 바둑을 뜻한다. *5 참고 | 정선은 두 사람 사이 다소 실력차이가 나서 실력이 다소 떨어지는 쪽이 흑으로 먼저 시작하는 바둑을 의미한다.

최신 AI 연구 흐름

"그렇다면, 현재 AI학계에서 핫(hot) 한 연구 결과물은 무엇인가?" 카카오AI리포트 편집진이 독자들부터 많이 받은 질문 중 하나입니다. 그래서 데이터, 의료, 공학에 전문성을 둔 주요 대학의 연구실로 부터 주목할만한 최신 논문을 추천받았습니다. 전문성이 높아, 조금은 딱딱할 수 있습니다. 그래도 그 안의 내용을 찬찬히 들여다 보면, 최신 연구 트렌드를 가늠할 수 있는 감(感)을 얻게 되실 겁니다. 조금은 벅차실 수도 있지만, 뿌듯할 결과를 그리며 시간을 내서 읽어 보실 것을 권해 드립니다. 최신 논문에 이어, 저명한 AI 학회인 ICML와 CVPR을 다녀온 카카오브레인의 안다비, 천영재 연구원 님의 참관기를 담았습니다.

최신 AI 논문 3선(選)

데이터, 의료, 공학 분야의 최신 논문이 각각 1편이 이번 장에 포함되어 있습니다. 첫 번째로, "텍스트의 주요 핵심 내용을 1~2문장 길이로 요약해 주는 기술"을 담고 있는 논문이 소개됩니다. 의료 관련 논문으로는 메탈 아티팩트(metal artifact)를 딥러닝을 활용해 제거하는 방법을 담고 있습니다. 메탈 아티팩트는 CT 영상 촬영 시, 촬영 대상자의 몸에 삽입된 금속성 물질로 인해 영상 내 발생하는 인공 음영을 의미합니다. 마지막 논문은 깊은 강화학습으로 대상을 인식하는 방식을 담고 있는 연구 결과물입니다.

데이터 고려대 데이터 사이언스&비즈니스 애널리틱 연구실: 빅데이터 및 사물인터넷 시대에 핵심적인 분석 방법론 개발 및 실제 사례 적용와 관련된 연구를 한다.

의료 서울대 방사선의학물리연구실: 방사선과 물질의 반응을 통한 에너지 전달과정에서 발생하는 흡수선량 계측과 계산(Radiation Dosinetry)에 중점을 두고 있다.

공학 서울대 인지지능연구실: 로보틱스(robotics)와 컴퓨터 비전(computer vision)을 연구한다.

글 | 김형석 hskim0263@korea.ac.kr

한때는 요리사가 꿈이었지만, 통계가 재미있어 결심한 대학원 석사과정 중 텍스트 마이닝 및 자연어 처리에 빠지게 되어 돌연 석박사 통합과정으로 전환하였다. 현재는 고려대학교 산업경영공학과 박사과정에 있으며, 주 연구 분야는 정보 검색(IR) 및 문서요약(Text Summarization)이다. 투박한 생김새와 다르게 디테일을 추구하는 섬세한 공대생. 언젠가는 매일 보는 논문을 잘 정리 요약 해주는 Tool을 만들어 대학원생들의 워너비스타를 꿈꾸기도 한다.

글 | 이지민 ljm861@gmail.com

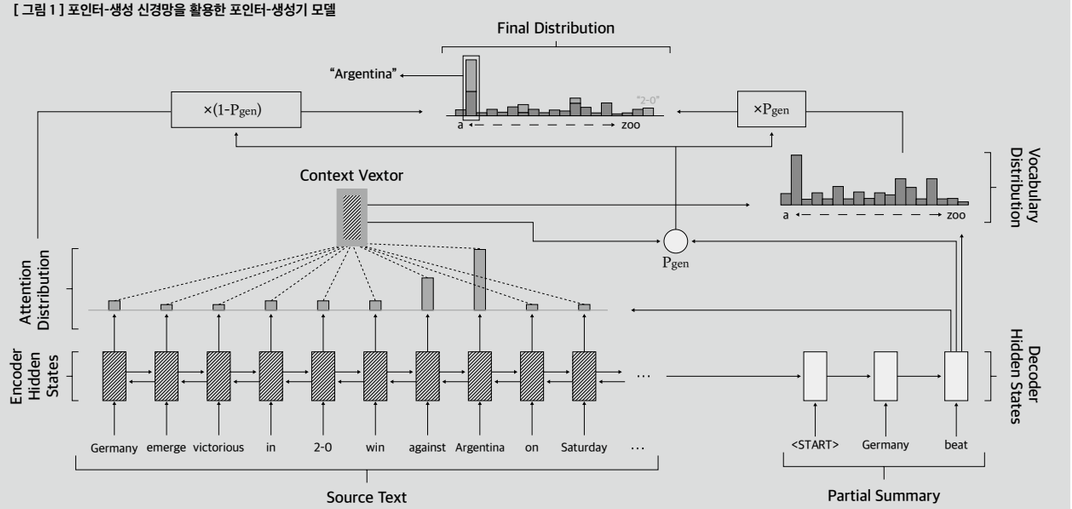
늘 행복하고 싶은 대학원생. 학부 때는 딥러닝과 전혀 상관없는 원자력 및 양자공학을 전공하였고, 대학원에 진학하여 의학물리학을 공부하던 중 교수님의 추천으로 딥러닝 공부도 함께 시작하게 되었다. 학부 때 여러 행사를 준비했던 경험이 있어 운이 좋게도 TensorFlowKR 운영진으로 활동하게 되었으며, 지식의 교류가 활발한 이 분야에 엄청난 매력을 느끼고 있다. 현재 라이언의 무심한 귀여움에 반해 짝사랑 중이며, 핸드폰 케이스, 마우스 패드를 포함한 라이언 굿즈를 10개 넘게 보유하고 있다.

글 | 이경재 kyungjae.lee@cpslab.snu.ac.kr

서울대 전기 컴퓨터 공학부를 졸업한 뒤, 동 대학원 석박사 통합과정으로 입학하였다. 현재는 박사과정에 있으며, 주 연구 분야는 모방학습과 지능형 로보틱스이다. 좀 더 세부적으로는 강화학습과 역강화 학습을 이용하여 로봇을 학습시키는 것이 목표이다. 한동안 군대 문제를 해결하기 위해 연구를 접고 영어공부에 매진했으나 영어실력은 그대로라고 한다. 최근 다시 연구를 시작하였으며 인공지능 및 로보틱스 분야의 많은 사람들과 교류하고 싶다. 석사과정 때는 이론 쪽 공부를 많이 하였으나 이제는 고속 주행 RC카나 매니퓰레이터와 같은 실제 로봇을 다뤄보려고 한다.

데이터 **요점을 말하다: 포인트 생성 네트워크 기반의 문서 요약 (Get To The Point: Summarization with Pointer-GENERATOR Networks)[*1]**

[그림 1] 포인터-생성 신경망을 활용한 포인터-생성기 모델



본 논문 구글 리서치(Google Research)팀이 ACL2017에서 발표한 연구 결과이다. 텍스트(text)로부터 주요 핵심 내용을 1~2문장 길이로 요약해 주는 기술의 소개가 이 논문의 골자이다. 기존 추상적 문서 요약(abstractive text summarization) 방식에서 문제시되던 부정확한 재현성과 반복적인 생성 문제를 각각 'Point-Generator Network'와 'Coverage Mechanism'을 통해 해결했다는 것이 이 연구의 특징이다.

딥러닝 이전 문서 요약의 경우, 규칙 위주(rule based)의 혹은 통계적 기반(statistical based) 방법론이 주를 이루었지만 상용화할 수 있는 부분이 제한적이었고 그 성능 또한 사람에 의한 문서 요약보다 월등히 떨어졌다. 하지만, 현재 RNN(Recurrent Neural Network), CNN(Convolutional Neural Network)과 같은 다양한 딥러닝 기법을 활용하게 되면서 문서 요약(text summarization)은 비약적으로 그 성능을 향상시킬 수 있었다.

문서 요약이란, 주어진 문서로부터 특정 사용자나 작업에 적합한 축약된 형태의 문서로 재생성하는 작업을 말한다. 이를 통해서 복잡도를 줄이면서 필요한 정보를 유지하는 것이 문서 요약의 주목적이다. 일반적으로 문서 요약의 접근법은 크게 2가지 방법으로 나누어 볼 수가 있다. 첫 번째는 추출식 접근법(extractive approach)으로 말 그대로 본문에서 의미 있는 부분을 선택하고 추출하여 이를 재정렬하도록 요약하는 접근법이다. 쉽게 말하면, 일종의 하이라이터(highlighter)라고 생각할 수 있다. 두 번째는 추상적 접근법(abstractive approach)이다. 이는 RNN과 같은 자연어 생성 기술을 활용하여 기존 텍스트(text) 문장을 토대로 창의적인 새로운 문장을 작성하는 방법이다.

이 논문의 연구 결과는 기본적으로는 두 번째 추상적 접근법을 취하면서, 필요 시에는 앞선 추출식 접근법을 하이브리드(hybrid)로 동시에 사용하여 좀더 자연스럽고 함축적인 문서 요약을 가능하게 한다.

시퀀스-투-시퀀스 주목 모델[*2]

라메쉬 날라파티(Ramesh Nallapati) 외 4명의 연구자가 2016년에 발표한 '시퀀스-투-시퀀스 순환신경망을 활용한 추상적 문서 요약'[*3] 논문의 주목 메커니즘(attention mechanism)을 통한 시퀀스 투 시퀀스 RNN 모델(seq2seq RNN model)[*4]을 바탕에 두고 있다. attentional seq2seq RNN model은 크게 4가지 모듈로 구성되어 있다.

1) 인코더 RNN(Encoder RNN)

원문 텍스트로부터 word2Vec으로 표현된 단어들을 각 단어(word-by-word) 단위로 읽는 모듈(module)이다. 일반적으로 인코더(encoder)는 양방향의 순서를 고려한 양방향 RNN(bidirectional RNN)을 활용한다.

2) 디코더 RNN(Decoder RNN)

요약을 구성하는 단어들의 시퀀스(sequence) 형태로 결과값

(output)을 반환하게 된다. 디코더 RNN(decoder RNN)은 인코더(encoder)와 다르게 한 방향 RNN(directional RNN)을 활용하여, 이전 요약 단계의 단어들을 입력값(input)으로 받게 된다. 원문 텍스트로부터 부분적인 문서 요약이 반환되는 것이다.

3) 주의 분포(Attention Distribution) 와 문맥 벡터(Context Vector)

원문 텍스트로부터 다음 단어를 생성할 때, 원문 텍스트(text)의 단어에 대한 확률인 주목 분포(attention distribution)는 인코더 RNN(encoder RNN)과 디코더 RNN(decoder RNN)의 은닉 상태(hidden state)를 입력값(input)으로 하여 아래와 같은 함수를 통해서 표현된다. 이는 직관적으로 해당 네트워크가 다음 단어를 생성 시 원문 텍스트 중에 어떠한 단어에 주목하여 봐야 할지를 표현해 주는 것이다.

[수식 1]

$$e_i^t = v^T tanh(W_h h_i + W_s s_t = b_{attn})$$

[수식 2]

$$a^t = softmax(e^t)$$

Context Vector

$$h_t^* = \sum_i a_i^t h_i$$

[수식 1, 2]에서의 v^T,W_h, W_s, b_{attn} 등은 네트워크의 학습 과정을 통해서 학습하는 매개변수(parameter)이다. 이러한 주목 분포(attention distribution)를 활용하여 인코더(encoder)의 은닉 상태(hidden state)와 가중합을 통해서 문맥 벡터(context vector)를 생성한다. 이는 디코더(decoder)가 해당 원문으로부터 어떠한 단어를 읽어 오는지를 표현한다..

4) 어휘 분포(Vocabulary Distribution)

최종적으로 문맥 벡터(context vector)와 디코더 은닉 상태(decoder hidden state)의 출력값(output)을 결합하여 2개의 선형 결합 층(layer)을 통해서 단어의 분포(vocabulary distribution)를 표현한다. 이는 원문 텍스트가 아닌 전체 단어(일반적으로 그 사이즈를 제한)에 대한 확률을 표현한다. 이를 통해서 최종적으로 문서 요약을 통해 생성될 단어들이 순서대로 표현된다.

포인터 생성 네트워크

본 논문에서 제안하는 '포인트 생성 네트워크(pointer-generator network)'는 기준 모델(baseline model)로 언급한 상기 '주목 기반 순환신경망 모델'(attentional seq2seq RNN model)과 2015년 비냘스(Vinyals) 외 2명의 연구자가 제안한 '포인터 네트워크(pointer network)'[5]의 중간에 있는 하이브리드 네트워크(hybrid network)로 간주될 수 있다. 상기 2개 모델의 하이브리드 네트워크 기능을 수행하는 'Pointer-generator network'를 통해 요약(summarization)단계에서, 기존 텍스트 본문에서의 단어들을 (1-P_{gen})의 확률로 인용하거나 새로운 단어(novel words)들을 P_{gen}확률로 생성할 수 있다. 아래 그림을 통해서 '포인트 생성 네트워크'의 다이어그램(diagram)을 확인할 수 있다.

> The generation probability

$$p_{gen} = \sigma(w_h^T h_t^* + w_s^T s_t + w_x^T x_t + b_{ptr})$$

$p_{gen} \in [0, 1]$, soft switch

> The probability distribution over the extended vocabulary

$$P(w) = p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} a_i^t$$

> if w is an out-of-vocabulary

P_{vocab} is zero

> if w does not appear in the source document

$\sum_{i:w_i=w} a_i^t$ is zero

직관적으로 '포인트 생성 신경망(pointer-generator network)'은 (1) 기존 baseline model의 출력값(vocabulary distribution)과 (2) 원문 텍스트로부터의 주목분포 사이에서 '포인터 생성 신경망' 기반의 학습된 생성 확률(P_{gen})을 반영하여 새로운 단어를 생성하여 문서를 요약(abstractive summarization)하거나, 기존 원문 텍스트를 재현하여 요약(extractive summarization)할지를 학습을 통해서 결정하는 역할을 수행하게 된다. 예를 들면, 생성 확률(P_{gen})이 0에 가까우면 기존 원문 단어로부터 재현된 단어들을 통해 문서 요약이 수행되고, 이와 반대로 생성 확률(P_{gen})이 1에 가까우면 encoder-decoder기반의 생성 단어들을 통해서 문서 요약이 수행되도록 하는 것이다.

Coverage mechanism

기존 주목 기반 순환신경망 모델의 문서 요약 단계에서 특정단어가 재활용되는 문제를 해결하기 위해서, 이 연구에서는 coverage vector를 통해서 일종의 패널티 조건(penalty term)을 손실(loss)에 반영하여 문제를 사전에 방지하고자 하였다. 이는 현재까지 사용되었던 단어의 주목 분포의 누적값을 coverage vector라 정의하고, 이를 'covLos'에 반영하여 요약 단계에서 같은 위치에 반복적으로 등장하는 것을 방지하는 역할을 한다. 이러한 'covLos'를 반영하여 기존 attention distribution과 학습을 위한 Loss는 기존 방법에서 아래와 같이 변화된다.

> To solve repetition problem, which is a common problem for seq2seq model

- $c^t = \sum_{t'=0}^{t-1} a^{t'}$

> Coverage Vector

- The sum of attention distribution over all previous decoder time steps
- Ct is (unnormailzed) distribution over the source document words
- C0 is zero vector

> Renewal of the attention mechanism

$$e_i^t = v^T tanh(W_h h_i + W_s s_t + w_c C_i^t + b_{attn})$$

$$a^t = softmax(e^t)$$

- To penalize the network for attending to same parts again.

$covloss_t = \sum_i min(a_i^t, c_i^t)$ $\quad$ $loss_t = -logP(w_t^*) + \gamma \sum_i min(a_i^t, c_i^t)$

- Extra loss term to penalize any overlap between the coverage vector and the new attention distribution

아래는 기존 baseline-model[6]과 본 논문의 'Summarization with Pointer-Generator Networks'의 문서 요약 예시이다.

baseline-model[6] 과 본 논문의 'Summarization with Pointer-Generator Networks'의 문서 요약 예시

Original Text (truncated): lagos, nigeria (cnn) a day after winning nige- ria's presidency, muhammadu buhari told cnn's christiane amanpour that he plans to aggressively fight corruption that has long plagued nigeria and go after the root of the nation's unrest. buhari said he'll "rapidly give attention" to curbing violence in the northeast part of nigeria, where the ter- rorist group boko haram operates. by cooperating with neighboring nations chad, cameroon and niger, he said his administration is confident it will be able to thwart criminals and others contributing to nigeria's instability. for the first time in nigeria's history, the opposition defeated the ruling party in democratic elections. buhari defeated incumbent goodluck jonathan by about 2 million votes, according to nigeria's independent national electoral commission. the win comes after a long history of military rule, coups and botched attempts at democracy in africa's most populous nation.

Baseline Seq2Seq + Attention: UNK UNK says his administration is confi- dent it will be able to destabilize nigeria's economy. UNK says his admin- istration is confident it will be able to thwart criminals and other nigerians. he says the country has long nigeria and nigeria's economy.

Pointer-Gen: muhammadu buhari says he plans to aggressively fight cor- ruption in the northeast part of nigeria. he says he'll "rapidly give at- tention" to curbing violence in the northeast part of nigeria. he says his administration is confident it will be able to thwart criminals.

Pointer-Gen + Coverage: muhammadu buhari says he plans to aggressively fight corruption that has long plagued nigeria. he says his administration is confident it will be able to thwart criminals. the win comes after a long his- tory of military rule, coups and botched attempts at democracy in africa's most populous nation.

두 번째로 제시된 baseline(seq2seq+attention)의 경우에서 'UNK'으로 표시된 비이상적인 단어들로 인해 부정확한 요약 결과가 생성되는 반면에, 해당 방법론을 사용한 경우 필요 시 원문 텍스트의 어휘들을 재현하면서도, 함축적인 새로운 어휘들을 통해서 문서 요약이 잘 수행된 것을 확인할 수 있다. 이로써 본 접근법이 'abstractive'와 'extractive'의 장점을 잘 반영한 문서 요약이 가능함을 알 수 있다.

논문의 한계점 및 시사점

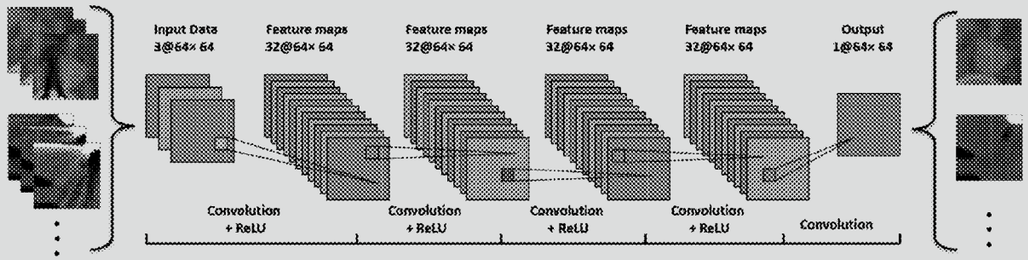
본 논문에서 소개된 연구 내용은 원문 텍스트와 문서 요약된 텍스트가 쌍으로 존재할 경우 수행이 가능한 접근법이다. 한글에서는 한 쌍의(pair) 학습 데이터 구성이 쉽지 않아, 정확한 학습이 어려울 것으로 예상되어 한글에서 연구 결과를 적용시키기는 어려울 것으로 판단된다. 한글로 된 뉴스의 첫 문단을 문서의 요약 문단으로 정의하고 학습하는 방법도 하나의 대안이 될 수 있을 것이라 생각된다.

또한 영문과 달리 조사 및 어미 등의 결합이 존재하고 있는 한글의 특성상 자연스러운 문장 요약이 매우 어려울 것으로 판단된다. 이러한 문제점을 잘 반영하여 함축적이고 가독성 좋은 문서 요약을 가능하도록 하는 후속 연구 과제는 유의할 것으로 평가된다.

의료

CNN 기반 엑스레이 CT 메탈아티팩트 제거 방법 (Convolutional Neural Network based Metal Artifact Reduction in X-ray Computed Tomography)[*7]

[그림 1] MAR에 사용되는 CNN 구조



영상(image) 기반 문제 해결에 CNN과 GAN과 같은 딥러닝 알고리듬이 높은 성능을 보이고 있는 요즘, 의료 AI 분야에서도 의료 영상 데이터를 이용한 연구가 활발히 진행되고 있다. 이번 호에서는 의료 영상 중 잘 알려진 CT (Computed Tomography) 영상의 여러 문제들을 딥러닝을 통해 해결해보고자 하는 논문에 대해 리뷰하고자 한다. 소개하는 논문은 CT 영상에서 발생하는 메탈 아티팩트(metal artifact)[*8]를 딥러닝을 이용해 제거해보고자 하는 연구를 담고 있다.

논문 소개

본 논문은 직접 CT 영상 데이터베이스를 만들어 CNN과 기존의 MAR(Metal Artifact Reduction)방법을 적절히 차용하여 향상된 성능을 보여준다. 데이터 확보를 위해 필요한 CT 영상을 수치 시뮬레이션(numerical simulation)하여 직접 생성한 것과, CNN 학습 시 기존에 알려진 MAR 방법인 BHC (Beam Hardening Correction)와 LI (Linear Interpolation)를 학습에 반영하고자 한 것이 흥미롭다.

연구 방법

1) Metal Artifact Database 생성(numerical simulation 방법 사용)

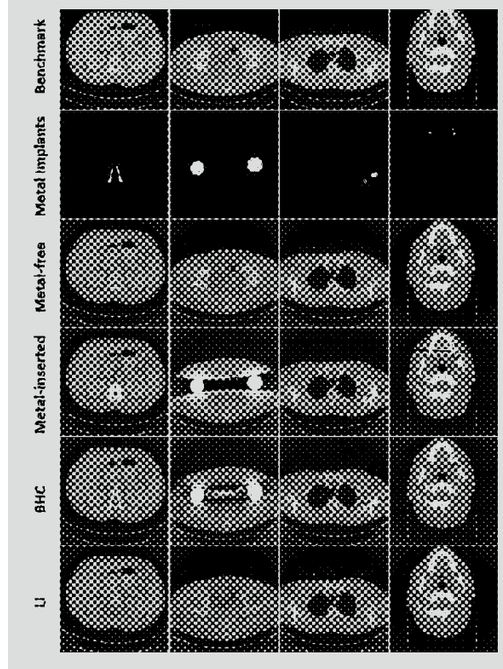
· "The 2016 Low-dose CT Grand Challenge" 학습 데이터셋 (training dataset)을 사용하였다.

· 위 데이터 셋에서 메탈 아티팩트가 발생한 영상을 추출하여 금속 부분을 분리(segmentation)하고 바이너리 이미지(binary image)로 저장하는 방법을 통해 [그림 2]의 Metal Implants 에 해당하는 총 15개의 금속 형태(metal shape)를 추출하였다.

· [그림 2]의 Metal-free에 해당하는 74장의 금속이 없는(metal-free) CT에서 추출한 metal shape를 최대한 실제 임상 케이스(clinical case)에 맞도록 금속의 모양, 위치, 재질(티타늄, 철, 구리, 금) 등을 조절하고 삽입하여, 총 100개의 CT 를 합성하였다. 이것이 [그림 2]의 Metal-inserted에 해당한다.

· Metal-inserted 영상을 기존에 사용되던 MAR 방법인 BHC과 LI를 적용하여 각각 추가 영상 [그림 2]의 BHC와 LI 를 획득하였다.

[그림 2] 유형별 대표 샘플



2) CNN 학습

· 입력 : 64×64×3 image patch (Metal-inserted, BHC, LI 영상을 3 채널로 묶고, 작은 패치(patch)로 추출, 원본 이미지 크기 : 512×512)

· 타겟 : 64×64 image patch (Input patch 와 동일 위치의 metal-free image 에서 추출한 patch)

· 총 10,000 patch samples 사용했으며 이 중 80%는 학습샘플(training samples)로, 20%는 검증샘플(validation samples)로 사용했다.

· 비용함수(cost function) : $H = \underset{H}{argmin} \frac{1}{R} \sum_{r=1}^{R} \| H(u_r) - v_r \|_F^2$ ($\| \cdot \|_F$: Frobenius norm, R : 학습샘플의 수, H(u_r): 입력 데이터셋, V_r: 타겟 데이터셋)

· 학습 소요 시간 : 25.5 시간(2000 iterations with GeForce GTX 970 GPU)

3) CNN-MAR method

· [그림 3] 에서 CNN을 접목한 MAR 방법을 도식화하여 보여주고 있다.

· 먼저 원본 사이노그램(Original Sinogram) 으로부터 Original(Metal-inserted)[*9], BHC, LI 영상을 생성하여 3 채널로 묶는다.

· 2)에서 학습된 CNN을 통해 3 채널로 묶인 영상으로부터 메탈 아티팩트가 감소된 CNN 생성 이미지(CNN Generated Image)가 생성된다.

· CNN Generated Image 로부터 tissue processing을 하여 CNN Prior 영상을 생성하고, 전방 투시법(Forward Projection)을 통해 1차 사이노그램(Prior Sinogram)이 생성된다.

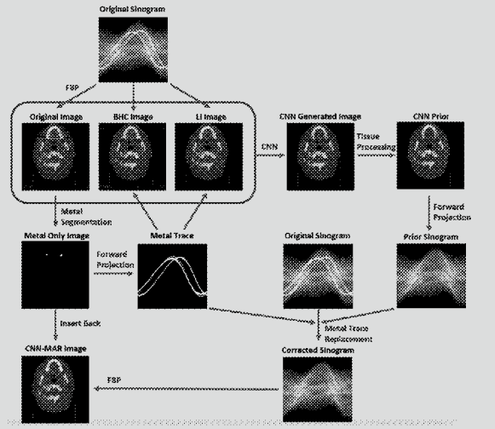
사이노그램이란?
CT는 방사선(x-ray)을 인체에 투과시켜 얻은 정보들로 영상을 재구성하여 만들어진다. x-ray가 어떤 물체를 통과하고 난 이후의 세기를 측정하면, p(s, Φ)를 획득할 수 있게 되는데, 이를 조사각도 Φ에서의 투사된 영상(projection)이라고 한다. 그리고 x-ray를 여러 각도에서 조사하여 투사된 영상을 얻으면, 각도에 따라 p(s, Φ)도 달라진다.[*10] 이렇게 획득한 p(s, Φ)를 s와 Φ를 양 축으로 한 2차원 평면에 나타낸 것이 사이노그램이다.[*11]

· 맨 왼쪽의 Original (Metal-inserted) Image 로부터 금속 부분만 분리하여 금속부분만 나타난 이미지(Metal Only Image)를 생성하고 전방투사법을 하면 금속 모양(Metal Trace) 만 있는 사이노그램이 생성된다.

· Metal Trace Sinogram 과 Prior Sinogram을 이용해 Original Sinogram 으로부터 메탈의 흔적을 제거하면(Metal Trace Replacement)를 하면 보정된 사이노그램(Corrected Sinogram)이 된다.

· Corrected Sinogram을 FBP (Filtered Back Projection: Sinogram을 영상으로 변환해주는 과정인 FBP(Filtered Back Projection))를 하면 최종적으로 CNN-MAR 이미지가 생성되게 된다.

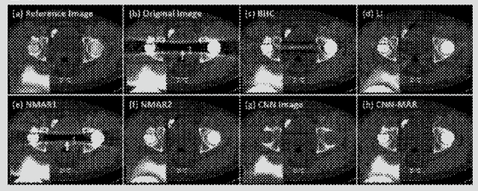
[그림 3] CNN-MAR 방법의 흐름도(flowchart)



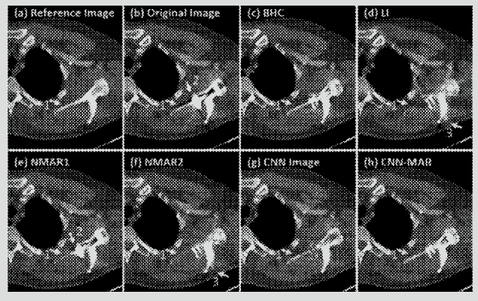
(4) Numerical Simulation을 통해 생성된 영상(Metal Artifact Database)에 대한 CNN-MAR 성능 평가

학습에 사용되지 않은 세 가지 케이스를 통해 제안된 CNN-MAR 방법의 성능을 평가해보았다.

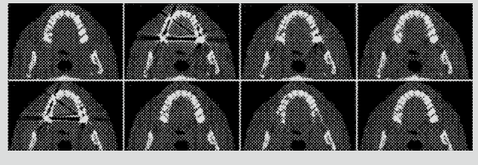
[그림 4] 고관절 인공보철물(two hip prostheses)이 삽입된 고관절 영상에서의 실험 결과, Case 1



[그림 5] 여러 보철물(two fixation screws and a round metal inserted in a bone)이 삽입된 어깨 부위 영상에서의 실험 결과, Case 2



[그림 6] 충치 치료 시 사용하는 치과용 충전제(several dental fillings)가 삽입된 치아 부위 영상에서의 실험 결과, Case 3

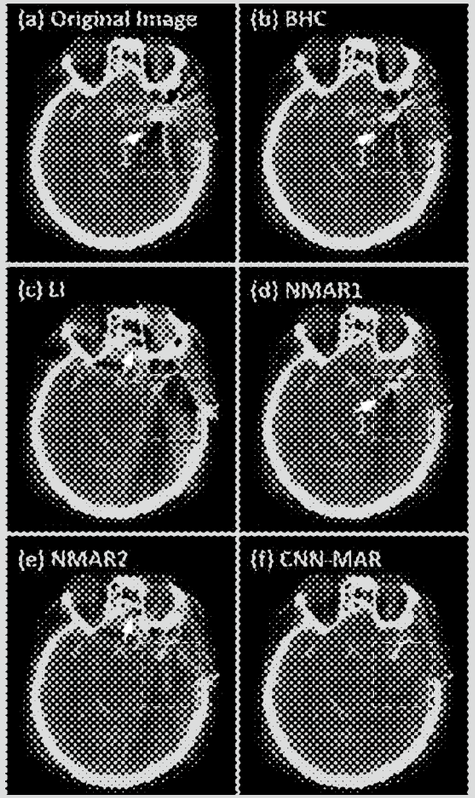


본 논문에서 제안한 CNN-MAR 방법을 비롯하여 기존에 사용되던 MAR 방법인 BHC, LI, NMAR (Normalized Metal Artifact Reduction) 방법을 통해 생성된 영상들과 레퍼런스 영상을 비교하기 위해서, 평균 제곱근오차(Root Mean Squared Error, RMSE) 와 구조유사도(Structural Similarity, SSIM) 인덱스(index)를 사용해 정량적으로 평가하였다.

5) 실제 환자 CT (Real Data) 에 대한 CNN-MAR 성능 평가

CNN-MAR 방법의 성능을 평가하기 위해 Numerical Simulation을 통해 생성된 영상이 아닌 실제 환자의 영상도 평가에 사용되었다.

[그림 7] 두경부 영상에서의 실험 결과



환자는 머리에 외과용 클립(surgical clip) 이 삽입되어 있으며 [그림 7]을 보면, 촬영에는 지멘스(Siemens)의 소마톰(SOMATOM Sensation 16 CT scanner)이 사용되었다.(120 kVp, 496 mAs) 촬영을 통해 1160개의 투사된 영상(projection view)을 얻었으며, 외과용 클립으로 인해 발생한 메탈 아티팩트를 CNN-MAR 및 다른 방법으로 개선한 뒤 그 성능을 비교하였다.

실험 결과

1) Numerical Simulation을 통해 생성한 영상에 대한 결과

[그림 4], [그림 5], [그림 6]을 보면 세 케이스 모두 CNN-MAR 방법을 적용했을 때, 피부 조직 부분을 잘 보존하면서도 가장 레퍼런스 이미지(reference image) 에 가까운 복원능력을 보여주었다.

[표 1] 수치 시뮬레이션 연구결과, 복원된 영상들의 평균 제곱근오차(RMSE).(단위 : HU)

	Original	BHC	LI	NMAR1	NMAR2	CNN	CNN-MAR
Case 1	155.0	86.3	46.2	121.2	35.4	33.1	29.1
Case 2	71.5	44.4	54.5	50.4	41.4	31.5	22.8
Case 3	320.3	183.5	107.3	234.9	82.3	83.4	58.4

[표 1]은 각각의 MAR 방법으로 복원된 영상들과 레퍼런스 이미지와의 RMSE(평균 제곱근오차)를 계산한 결과이다. CNN-MAR 방법을 통해 복원된 영상이 모든 케이스에서 가장 작은 RMSE 값을 가짐을 알 수 있다.

[표 2] 수치 시뮬레이션 연구 결과, 복원된 영상들의 구조 유사도(SSIM)

	Original	BHC	LI	NMAR1	NMAR2	CNN	CNN-MAR
Case 1	0.565	0.576	0.930	0.887	0.935	0.940	0.943
Case 2	0.883	0.854	0.931	0.955	0.950	0.965	0.977
Case 3	0.522	0.536	0.886	0.833	0.942	0.932	0.967

[표 2]는 각각의 MAR 방법으로 복원된 영상들의 구조 유사도를 비교한 결과이다. 구조 유사도 역시 CNN-MAR 방법을 통해 복원된 영상에서 가장 높은 결과를 보였다.

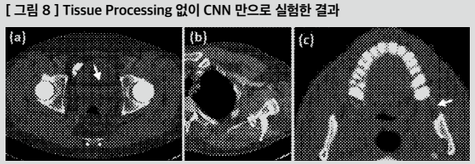
2) 임상 적용(clinical application)

[그림 7]을 보면, 실제 환자 CT 영상에 대해서도 CNN-MAR 기법이 가장 우수한 성능을 보인 것을 확인할 수 있다. CNN-MAR 방법을 적용할 경우, 메탈 아티팩트가 잘 제거되었고 (화살표 1로 표시된 검은 부분들이 잘 복원됨), 뼈 구조(Bony structures) 역시 변형시키지 않았다.

3) Properties of the Proposed CNN-MAR

CNN-MAR 방법의 특성을 알아보기 위해 다양한 실험을 추가적으로 진행하였다.

[그림 8] Tissue Processing 없이 CNN 만으로 실험한 결과



· 우선 tissue processing 의 효과를 확인하기 위해, tissue processing 없이 CNN 만으로 prior image 를 생성해본 결과, 메탈 아티팩트가 완전히 사라지지 않고 일부 남아있는 모습을 보였다. ([그림 8]의 화살표 참고)

· 따라서 CNN 과 tissue processing 은 서로 상호보완적인 관계로 활용되고 있다는 것을 알 수 있다.

· 또한 CNN의 입력값(input)으로 원본 이미지(Original Image)와 함께 들어가는 기존의 다양한 MAR 적용 영상들에 대해서도 평가해보고자, 입력값(input)을 다양한 채널로 구성하여 추가 실험을 진행했다. (2 채널 : Original + LI / 3 채널 : Original + BHC + LI / 5 채널 : Original + BHC + LI + NMAR1 + NMAR2)

[표 3] 채널별 CNN과 CNN-MAR방식으로 복원된 영상들의 평균 제곱근오차 (RMSE)와 구조 유사도(SSIM)

		Two-channel input image		Three-channel input image		Five-channel input image	
		CNN image	CNN-MAR	CNN image	CNN-MAR	CNN image	CNN-MAR
Case 1	RMSE	40.0	31.0	33.1	29.1	27.3	27.7
	SSIM	0.931	0.938	0.940	0.943	0.942	0.942
Case 2	RMSE	43.4	35.9	31.5	22.8	26.9	22.4
	SSIM	0.956	0.971	0.965	0.977	0.968	0.975
Case 3	RMSE	97.3	66.0	83.4	58.4	79.5	59.3
	SSIM	0.912	0.956	0.932	0.967	0.954	0.971

· [표 3]의 결과를 보면 5 채널 의 경우 성능이 가장 좋았으며 [표 3], 이는 CNN이 기존의 다양한 종류의 MAR 방법들을 통합할 수 있는 가능성을 지닌다고 저자들은 설명하였다.

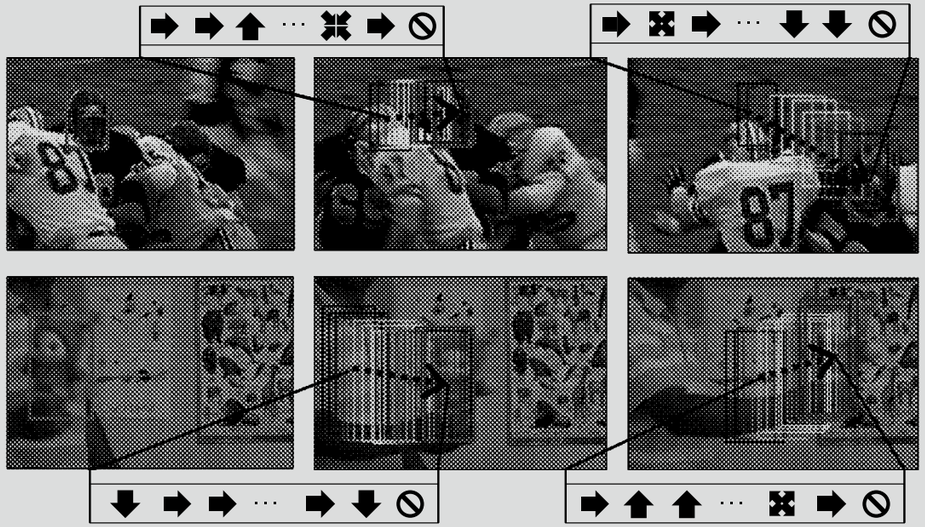
결론

연구 결과, 제안된 CNN-MAR 방법에서 사용된 CNN 과 tissue processing 은 상호 보완적인 관계를 가지며, 함께 사용되어 메탈 아티팩트를 줄이고 주변의 세밀한 구조들을 복원하는 우수한 성능을 보여주었다. (CNN 은 서로 다른 MAR 방법으로부터 유용한 정보를 융합하였으며, tissue processing 은 대부분의 아티팩트를 제거하고 우수한 품질의 prior image를 생성하였다.) 이후 추가 연구를 통해 학습데이터를 늘리고 다양한 MAR 방법을 융합하여, CNN-MAR 방법의 성능을 향상시킬 예정이라고 한다.

공학

깊은 강화학습을 통한 시각적 추적을 위한 행동-결정 네트워크 (Action-Decision Networks for Visual Tracking with Deep Reinforcement Learning)[*12]

[그림 1] 연속적 액션을 통한 비주얼 트래킹의 개념



이 논문은 2017년 컴퓨터 비전 및 패턴인식학회(Conference on computer Vision and Pattern Recognition, CVPR)에 발표된 논문으로 컴퓨터 비전 분야에서 중요하게 다루는 객체 추적 문제(object tracking)에 깊은강화학습(deep reinforcement learning)을 다루고 있다. 현재 객체 추적 문제를 해결하는 최신 기술(state-of-the-art)은 합성곱인공신경망(convolutional neural network, CNN)을 기반으로 물체를 탐지(object detection)해서 목표로 하는 객체와 배경을 구별하는 방법을 사용하는 것이다[*13]. 이러한 방법들은 다음과 같은 문제가 있었다.

- 객체 위치 후보를 전부 탐색하는 비효율성
- 객체 테두리상자 라벨(label)의 필요성

논문에서는 깊은강화학습을 이용한 ADNet(Action-Decision Network)을 도입하여 문제들을 해결하였다. 객체 추적 문제를 마르코프 결정 과정(Markov Decision Process, MDP)으로 생각하여 이전 프레임에서 객체의 테두리 상자(bounding box)가 액션을 통해 움직여서 현재 프레임의 테두리 상자의 위치와 모양을 결정하도록 했다.

- 상태(state)는 현재 프레임에서 테두리 상자의 위치와 이전 10번의 액션으로 결정됨
- 액션은 상하좌우 움직임, 그 2배의 움직임, 테두리 상자의 크기 확대와 축소, 그리고 멈춤으로 구성됨
- 보상(reward)은 멈춤 액션을 했을 때 테두리 상자가 정답(ground truth)과 일치하는 정도가 높으면 1 아니면 -1을 가짐. ADNet은 현재의 상태가 입력값으로 들어오면 출력값으로 액션과 신뢰도(confidence)를 도출한다.

[그림 1]과 같이 이전 프레임의 객체 위치에서 시작하여 여러 번의 액션을 통해 현재 프레임에서 객체의 위치를 찾아간다.

ADNet의 학습은 다음과 같은 3단계로 나뉜다.

ADNet의 학습 3단계

지도학습(supervised learning)

· 상태를 입력으로 주고 액션과 신뢰도를 출력으로 하는 VGG-M[*14] 기반 CNN을 학습한다.
· 각 프레임의 정답을 기준으로 가우시안 노이즈(Gaussian noise)로 생성한 테두리상자들을 데이터로 사용하였다.
· 생성된 상자들이 정답에 가까워지기 위해 해야 하는 액션을 정하는 것이 네트워크 학습의 목표이다.
· 또한 이 상자의 내용물이 객체인지 아닌지 판별하는 것, 즉 신뢰도를 학습하는 것도 네트워크의 또 다른 목표이다.

깊은강화학습(deep reinforcement learning)

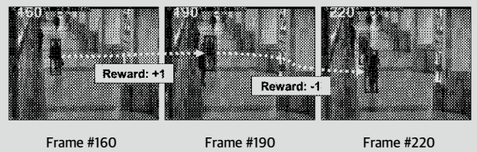
· Policy gradient 방법을 사용하여 네트워크에서 직접 정책(policy)을 얻는다.
· 지도학습에서 미리 학습된 네트워크를 정책 네트워크(policy network)의 시작점으로 활용한다.
· 한 프레임내의 마지막 예측지점이 객체와 맞냐 안맞냐를 통해 보상을 1 또는 -1로 하고 이를 바탕으로 정책 네트워크를 업데이트한다.

실시간 적응(online adaptation)

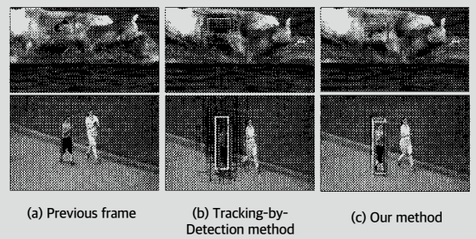
· 프레임에서 예측된 테두리상자를 정답의 기준으로 하는 샘플 데이터를 모아 실시간으로 지도학습을하여 성능을 향상시킨다.
· 신뢰도가 0.5 이하가 되어 객체를 놓쳤다고 판단되면 현재 위치에서 가우시안 노이즈로 생성한 주변 위치 중에서 가장 신뢰도가 높은 지점을 새로 위치로 지정하는 re-detection 과정이 있다.

강화학습을 사용하면 중간중간에 객체 위치 라벨이 없는 데이터에 대해서도 앞뒤 프레임의 정보를 활용하여 보상을 설정해 주는 식으로 학습할 수 있다[그림 2]. 따라서 ADNet은 데이터가 불완전한 준지도학습(semi-supervised learning)에도 대응할 수 있다.

[그림 2] 준지도학습(semi-supervised learning) 사물 추적 시뮬레이션



[그림 3] 탐색 전략 비교



[그림3]에서 ADNet을 활용한 객체 추적은 훨씬 적은 양의 탐색이 필요한 것을 알 수 있다. 이를 통해 최첨단기술 대비 약 3배 빠른 속도 향상을 보여 주었다.

- 최첨단기술인 (b)의 경우 물체 탐지(object detection)를 기반으로 하기 때문에 주변의 여러 후보 영역들을 탐색해야 한다. 현재 객체 추적 문제를 해결하는 CNN기반의 방법의 경우 프레임당 256개의 영역이다.
- ADNet의 경우 액션을 통해 이동하는 영역만 탐색하면 된다. 평균적으로 프레임당 28.26개의 영역이다.

[표 1] 사용자 반응 수집 지연에 따른 상태 CTR 변화

	Aigorithm	Prec.(20px)	IOU(AUC)	FPS	GPU
Non real - time	ADNet	88.0%	0.646	2.9	O
	ADNet-fast	85.1%	0.635	15.0	O
	MDNet [24]	90.9%	0.678	< 1	O
	C-COT [9]	90.3%	0.673	< 1	O
	DeepSRDCF [8]	85.1%	0.635	< 1	O
Real - time	HDT [25]	84.8%	0.564	5.8	O
	MUSTer [15]	76.7%	0.528	3.9	X
	MEEM [42]	77.1%	0.528	19.5	X
	SCT [5]	76.8%	0.533	40.0	X
	KCF [13]	69.7%	0.479	223	X
	DSST [7]	69.3%	0.520	25.4	X
	GOTURN [12]	56.5%	0.425	125	O

위 표는 ADNet의 결과와 다른 알고리듬들을 비교한 것이다. Prec.(20px)은 정답과 예측한 테두리 상자의 중심간 거리가 20픽셀 이하인 비율이고 IOU(AUC)는 두 상자의 겹치는 비율을 뜻한다. ADNet은 비실시간 최첨단 기술에 근접한 성능을 내면서도 속도는 더욱 빠른 결과를 보인다. 또한 실시간 적응 단계에서 학습에 필요한 샘플 데이터 수를 비교적 적게 뽑는 ADNet-fast의 경우 FPS 15.0으로 실시간으로 작동하는 알고리듬 중 최고 성능을 보인다.

*1 출처 | See, A. (2017). Get to the point : Summarization with pointer-generator networks, doi : arXiv:1704.04368. *2 참고 | sequence-to-sequence attentional model *3 참고 | abstractive text summarization using sequence-to-sequence RNNs and beyond *4 참고 | 부정확한 재현성과 반복적 단어의 재활용 등이 문제점으로 지적된다. *5 참고 | Vinyas, O., Fortunato, M., Jaitly, N. (2017). Pointer networks. doi : arXiv:1506.03134 *6 논문 | Nallapati, R., Zhou, B., & Santos C. (2016). Abstractive text summarization using sequence-to-sequece RNNs and Beyond, doi : arXiv:1602.06023 *7 논문 | Zhang, Y. & Yu, H. (2017). Convolutional neural network based metal artifact reduction in X-ray computed tomography, arXiv:1709.01581. *8 참고 | CT 영상은 몸에 방사선(x-ray)을 조사하여 획득하게 되는데, 체내에 금속성 물질(예: 금니, 인공 관절, 인공심장박동기 등)을 삽입한 환자가 CT를 찍게 되면 Beam Hardening Effect, 산란(scatter), 푸아송 노이즈(Poisson noise), 환자의 움직임(motion) 등의 다양한 원인에 의해 영상에 인공음영이 생성된다. 이렇게 몸에 삽입된 금속성 물질로 인해 발생하는 인공음영을 메탈 아티팩트라고 일컫는다. 메탈 아티팩트를 감소시킬 목적으로 지멘스(SIEMENS)의 iMAR#, 필립스(PHILIPS)의 OMAR, GE헬스케어(GE Healthcare)의 Smart MAR 등의 상용화된 MAR 소프트웨어가 출시되었지만, 아직까지도 촬영 부위에 따라 효과적이지 못한 경우가 많이 발생하고 있다. 따라서 이 문제를 해결하고 영상의 품질을 개선하려는 연구들이 현재까지 계속되고 있다. 최근 딥러닝을 활용한 영상연구가 괄목하게 발전하면서, 메탈 아티팩트 문제도 딥러닝을 접목해서 해결하려는 연구가 활발하게 전개되고 있다. *9 참고 | 같은 논문의 III. B. Tissue Processing에 자세히 설명되어 있습니다. *10 참고 | http://terms.naver.com/entry.nhn?docId=3572221&cid=58944&categoryId=58970 *11 참고 | https://patentimages.storage.googleapis.com/US8509504B2/US08509504-20130813-D00000.png *12 참고 | http://openaccess.thecvf.com/content_cvpr_2017/papers/Yun_Action-Decision_Networks_for_CVPR_2017_paper.pdf *13 논문 | Nam, H. & Han, B. (2016). Learning multi-domain convolutional neural networks for visual tracking. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. *14 논문 | Chatfield, Ken, et al. (2014). Return of the devil in the details : Delving deep into convolutional nets, doi : arXiv:1405.3531.

최신 기계학습의 연구 방향을 마주하다

ICML 2017 참관기

호주 시드니에 있는 달링하버(Darling Harbour)는 정말 아름다운 항구다. 올 8월에 운 좋게도 여기를 다시 오게 되었다. 이번엔 놀러 온 게 아니고 말하자면 일하러…….

이렇게 멋진 곳에서 새로운 연구 접하며 굳은(?) 뇌를 말랑말랑하게 자극하고, 바다를 보며 혼자 고심도 해보는 시간을 가질 수 있어 행복했다. 세계에서 모여드는 대단한 연구자들을 한 자리에서 만날 수 있음도 물론이지만, 국제기계학습학술대회(International Conference on Machine Learning, ICML)는 올해 34회를 맞은, 세계에서 가장 명성 있는 머신러닝 학회 중 하나이고 다양한 세부 분야를 다루므로 연구의 시야를 넓히기에 제격이었다.

특히 'Lifelong learning'[1] 이나, 'Interpretability'[2] 등은 눈여겨볼 만했다. 최근 몇 년간 딥러닝이 머신러닝 주요 방법론 측면에서 가장 높은 지위에 올랐지만, 여전히 확률모델 기반의 방법론과 강화학습도 활발히 연구되고 있다. 더 중요한 사실은 연구 방법론들이 따로 발전하는 게 아니라 서로 경계를 넘나들며 조합되고 있다는 것이다. 이번 학회에서 확률모델과 딥뉴럴 네트워크(Deep Neural Network, DNN)을 조합한 LDA(Latent Dirichlet Allocation) 모델도 보았고, 딥뉴럴 네트워크에 강화학습을 적용하여 성능을 높인 사례도 보았다.

응용분야 별로 열린 워크숍에서는 자율주행 분야의 인기가 많았는데, 오전에 학회장에 들어가려고 했더니 내부 공간에 사람이 꽉 차서 출입문 앞에서 들어가지 못하는 일까지 겪어야 했다. (결국 오후에는 들어갔지만.) 뮤직 디스커버리 워크숍에도 참여했는데, 원하는 아티스트 스타일로 기타 연주 생성하기, 음악 스트리밍 서비스를 하는 글로벌 기업인 스포티파이(Sportify)의 음악 추천 방법 등 흥미로운 주제들이 많았다. 발표된 연구들 중 절반 정도는 전통적인 머신러닝 방법을 활용하였는데, 딥뉴럴 네트워크를 활용하여 조금 더 확장해 볼 만한 여지가 보였다.

글 | 안다비 avin.hero@kakaobrain.com

사람들은 '다비'라는 이름을 두고 단지 예쁜 한글이름이라고 생각하지만 실은 한자로 많다 '다', 크다 '비'로 많고 크다는 뜻을 갖고 있다. 주어도 없이 많고 크다니.. 참 욕심많은 이름이다. 그래서인지 매사에 의욕이 많지만 가끔은 생각이 많은 스스로에게 힘들어한다. 컴퓨터 사이언스를 배우고 연구하기를 6년, 그리고 엔지니어로 5년. AI 같은 기술적인 것을 원래 좋아하던 차에 알파고가 세상을 놀래키면서 AI가 핫하게 떠오르기까지 하니 시대를 참 잘 타고났다고 느끼는 요즘이다. 구글 검색, 넷플릭스 처럼 멋진 기술이 접목되면서도 동시에 지구상의 많은 이들이 좋아하는 유명한 서비스를 만들고싶은 자칭 야망가이자 속세 엔지니어이다.

RNN 관련 두 연구 소개

최근 언어 발화(speech)나 음악(music), 자연어 처리(natural language processing, NLP)에 관심을 가지게 되면서 시퀀스 형태의 데이터를 다뤄야 할 일이 많아졌다. 따라서 이에 적합한 모델인 순환신경망(recurrent neural networks, RNNs)에 많은 관심을 갖고 있었고, 여러 날에 걸쳐서 열린 4개의 RNN 세션에 모두 참여했다. 앞 두 세션에서는 RNN이라는 큰 틀에서, 풀고자 하는 문제에 최적화된 변형 모델을 제시하는 연구들이 주를 이루었다. 나머지 두 세션에서는 RNN을 현실적인 문제에 응용한 연구들로 구성되었는데 음악 및 동영상을 다루는 영역에서 RNN을 활용한 연구들이 많이 소개되었다.

앞의 두 세션에서 두 가지 연구가 기억에 남아 하나씩 구체적으로 얘기해 보고자 한다. 바로 RHNs(Recurrent Highway Networks)과 시퀀스 튜터(Sequence Tutor)인데, 두 연구 모두 여러 도메인에서 범용적으로 적용해 볼 수 있는 모델 혹은 방법론을 제시해 주기 때문에 도움이 될 것 같다.

Recurrent Highway Networks

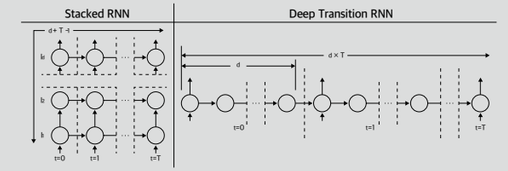
RHNs(Recurrent Highway Networks)를 이해하려면 먼저 Highway Networks(ICML, 2015[3])를 알고 있어야 한다. highway는 '고속도로' 라는 뜻을 가지고 있다. 간단히 말해 highway networks는 어떤 레이어를 통과할 때 해당 레이어에서 수행되어야 하는 선형 연산과 활성화(activation) 같은 연산을 거치지 않고 빠르게 지나만 가는 우회로도 제공하자는 것이다. 이렇게 함으로써 얻을 수 있는 효과는 '빠른' 학습이다.

보통은 레이어가 많아질수록 앞쪽 레이어에서는 그라디언트(gradient) 소실 문제로 인해 학습이 매우 느리거나 되지 않는 경우가 많은데, highway networks에서는 레이어를 연산 없이 통과하므로 레이어를 적게 사용하는 것처럼 동작하게 해준다. 레이어를 얼마만큼 거쳐 갈 것인지, 혹은 얼마나 연산 없이 그냥 지나칠 것인지를 학습하게 되는데 각각의 비중을 조정하기 위해 변형 게이트(transform gate)와 캐리 게이트(carry gate) 같은 2개의 게이트가 사용된다.

Highway Networks가 우리에게 주는 의의는 두 가지다. 하나는, 레이어가 아무리 많아져도 학습이 느려지지 않게 하는 모델을 만들 수 있다는 것이다. 다른 하나는, 모델의 하이퍼파라미터 중 하나인 레이어의 개수를 데이터의 복잡도에 관계없이 크게 잡아도 된다는 것이다. 데이터로부터 배우는 데 필요한 학습은 앞쪽 레이어에서부터 주로 일어나고 뒤쪽 나머지 레이어들에서는 그냥 highway를 통과하는 형태로 스스로 학습될 것이기 때문이다.

RHNs는DT-RNN(Deep Transition RNN) 은닉 상태(hidden state)의 전이에 highway network를 적용하여 학습 속도를 높인 모델이다. DT-RNN는 은닉 상태의 전이가 한 타임스텝에서도 여러 번 일어나는 RNN이다[4]. 이 연구에서 밝힌 사실은, 같은 수의 파라미터를 가지는 기존의 DT-RNN에 비해 RHN에서 학습 속도가 빨라졌으며 특히 은닉 상태 전이 부분의 여러 highway block 중 첫 번째 블록에서 가장 은닉 상태의 변화가 많고 나머지 블록들에서는 조금씩 정제만 하는 경향이 있었다는 것이다. 데이터의 복잡성이 크지 않을 때는 일반 (stacked) RNN처럼 매 타임 스텝마다 한 번씩만 전이해도 은닉 상태의 변화를 담아내는 데에 큰 무리는 없다는 의미로도 이해할 수 있다. RHN은 시퀀스 형태의 데이터를 다루는 문제에서 stacked RNN을 넘어서 적용해 볼 여지가 있는 모델이라고 생각한다.

[그림 1] Stacked RNN과 Deep Transition RNN의 도식화[5]



Sequence Tutor

Sequence Tutor는 구글 마젠타(Magenta)팀에서 주축이 되어 진행한 연구이다. 연구 내용을 한마디로 요약하자면, 작곡과 같은 시퀀스 생성(sequence generation) 문제에서 모델을 RNN으로 학습시킨 뒤 부족한 부분을 강화학습을 통해 정제하자는 것이다.

여기서 말하는 부족한 부분이란, 길이가 상당히 긴 시퀀스를 학습시켰을 때 전체적인 일관성이 깨지는 문제이다. 보통 노래는 처음부터 끝까지 전체적인 멜로디의 규칙성이나 스타일이 유지되어야 하는데, 이런 일관성을 유지하게끔 RNN을 학습시키는 것은 쉽지 않다. 다시 말해, 듣기에 별로 감동스럽지 않은 멜로디를 만들어 낸다. 이는 Character RNN이 얼핏 보기에 그럴듯한 문장을 생성하지만, 실제 사람이 보기에 맥락적으로 '말이 되는' 문장을 생성하지는 못하는 것과 같다.

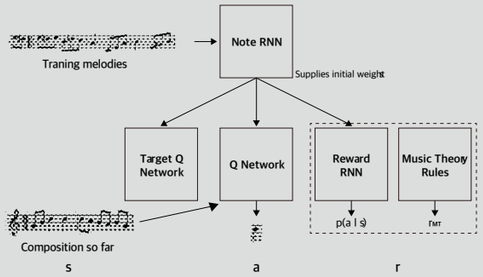
Sequence Tutor에서는 말이 안 되는 노래를 말이 되게끔 하는 데에 강화학습이 들어간다. 방법론은 다음과 같다. 우선 LSTM-RNN으로 앞서 말한 '말이 안 되게' 작곡하는 모델을 만든다(Note RNN). 그런 뒤 이제 이 작곡 문제를 강화학습이라고 보고 현재까지의 작곡된 것은 환경(environment), 다음 멜로디를 행위(action)이라고 본다. 그러면 다음 행위를 추측하기 위한 정책

네트워크(policy network)를 구성해야 할 것이고, 이를 위해 가치 함수(value function)도 정의해야 할 것이다.

정책 네트워크에서는 DQN(Deep Q-Networks)을 사용하는데 Note RNN에서 학습된 것을 초기 모델로 사용한다. 여기서 핵심이 되는 부분은 가치 함수인데, 여기에 말이 안 되는 노래를 말이 되게끔 바꾸어 주는 음악이론 규칙이 들어간다. 예를 들면, 노래가 C장조라면 그 외 다른 음은 배제한다, 같은 음을 4번 이상 만들지 않는다 등 정의된 규칙에 맞는 음에 높은 보상(reward)을, 그렇지 않은 음에는 낮은 보상을 준다. 한편 정의된 규칙만을 따르게 하면 창의적이지 못한 경직된 멜로디만 생성될 것이므로 어느 정도의 창의성도 필요한데, 기존에 학습된 Note RNN으로부터의 복제 모델인 Reward RNN으로부터 끌어온다. 즉, 기존 음악에서는 어떤 음을 주로 사용했는지 힌트를 얻어 오는 것이다. 결과적으로 가치 함수는 Reward RNN에서의 보상과 음악이론 규칙에서의 보상을 더한 값으로 정의한다.

RNN에 강화학습을 접목한 이 방법론은 마법처럼 말이 되는 노래를 만들어 낸다. 사용한 데이터가 무엇인지, 음악 이론 규칙을 어떻게 정의했는지에 따라 다른 노래를 만들어 내겠지만 어쨌든 좀 더 그럴싸한 창작이 가능하게 된 것이다. 여기서는 주로 음악을 예로 설명했지만 시퀀스 생성이 필요한 다른 도메인에서도 시도해 볼 만한 방법론이라고 생각한다.

[그림 2] Sequence Tutor의 전체 구조도[*6]



인공지능이 바꾼 연구 생태계

얘기가 너무 연구 내용으로 빠진 것 같으니 다시 학회 분위기로 돌아가 보자. 최근 이 분야의 IT 업계의 높은 관심에 걸맞게 딥마인드, 구글, 페이스북, 바이두 등 글로벌 기업들의 연구 성과가 특히 눈에 띄었다. 체감상 절반 이상은 기업에서 이루어진 연구로 보였다. 장기간의 투자로 결과들도 굉장히 훌륭해서 그저 부러울 따름이었다. 딥뉴럴 네트워크은 수 년 전까지만 해도 캐나다의 소수의 연구자들에 의해 소소하지만 꾸준히 발전해 왔다.

그러던 어느 날 딥뉴럴 네트워크의 가능성이 입증되면서 이제는 기업에서도 적극적으로 인공지능 연구에 동참하게 된 것이다. 이를 학회장에서도 온전히 체감할 수 있었다.

그동안 학계와 산업계에서 꽤 명확하게 구별하여 연구돼 오던 다른 분야의 성장 양상과는 전혀 다르게 이 분야는 학계와 산업계에서 인기를 동시에 지속하며 폭발적으로 성장하고 있다. 머신러닝 연구 자체를 지향점으로 삼고 연구 결과를 통해 기여하고자 하는 리서치 사이언티스트와, 좋은 연구 결과를 잘 활용하여 현실 세계에서 도움이 될 만한 프로덕트를 만드는 데 관심 있는 리서치 엔지니어가 함께 어우러져 발전해 나가는 모습이 참 조화롭다는 생각이 든다.

마치며

현재의 인공지능 열풍이 지속될까? 앞으로는 사람이 인공지능이 사용되었다는 것을 인지하지 않을 정도로 대부분의 분야에서 인공지능을 활용하는 시대가 되지 않을까? 향후 10년 동안의 관전 포인트다. 개인적으로는 지금과 같은 관심이 지속되어서 10년 내에는 영화 '설국열차'에 등장했던 동시통역기가 현실화되고 더 이상 영어공부를 하지 않아도 되는 세상이 오길 매우 기대해 본다.

***1** 참고 | 태스크마다 별도의 학습데이터와 모델을 사용하는 것이 아니라 과거 태스크로부터 배운 지식을 현재의 태스크에 활용하자는 패러다임 ***2** 참고 | 블랙박스처럼 좀처럼 속을 알기 어려운 딥뉴럴 네트워크의 내부 동작을 설명하는 분야 ***3** 참고 | Srivastava, R.K., Greff, S., Schmidhuber, J., (2015). Highway Networks. doi : arXiv:1505.00387. ***4** 참고 | 사실 은닉 상태가 딱 한 번만 전이하라는 법은 없으니 조금 더 여러 번 전이할 수 있도록 만든 일반화된 RNN이라고 생각할 수 있다. ***5** 참고 | https://github.com/julian121266/RecurrentHighwayNetworks ***6** 논문 | Jaques, N. et al. (2017). Sequence Tutor: Conservative Fine-Tuning of Sequence Generation Models with KL-control, ICML.

2013년과 2017년의 CVPR을 비교하다

출발은 한두 주 전쯤으로 기억한다. 지인의 초대로 페이스북 그룹채팅방에 들어가보니 직접적으로 혹은 한다리 건너 알만한 대학원 연구실 학생들이며 회사에 소속된 연구원들 20~30 명이 모여 있었다. 채팅방 이름은 'CVPR2017_kor 수다방' 제목 그대로 CVPR에 참석하는 한국인들의 수다방이었다. 학회 전까지는 자기소개며, 어떤 비행기를 타고 가는지 등의 잡담을, 학회 중에는 실시간 발표자에 대한 평이나 저녁 시간 번개를 모집하는 글 등 자유로운 글들이 여과없이 올라왔다. 직감적으로 이번 학회에 상당히 많은 한국인들이 참석하게 될 것을 느꼈고 예상대로 인천공항과 기내에서 여러 명의 지인과 마주쳤다

컴퓨터 비전 및 패턴 인식 분야에는 '컴퓨터 비전 및 패턴 인식 컨퍼런스(CVPR)', '유럽컴퓨터비전학회(ECCV)', '국제컴퓨터비전학회(ICCV)' 등 외에도 다양한 학회가 있다. 그 중 CVPR은 규모나 영향력 지수(impact factor) 측면에서 가장 인기 있는 학회이다. 특히나 최근에는 딥러닝의 인기로 모든 지표에서 학회의 성장을 확인할 수 있다. 당장 작년과 비교해 봐도 논문 투고 수 40% 증가, 등록 인원 37% 증가, 스폰서 펀딩 금액 79% 증가 등 가파른 성장세를 보이고 있다. 개인적으로 가장 인상 깊었던 것은 기업체 지원(sponsor) 부분이었다. 2010년 전후만 해도 시그라프(SIGGRAPH) 학회에 참석하게 되면 스폰서 리스트에 표시된 수십 개의 기업체 마크와 화려한 전시 부스를 보며 내심 부러웠는데, 이제 CVPR은 시그라프보다 많은 130여개 업체가 스폰서십으로 참여할 정도로 기업체로부터 대단히 큰 관심과 지원을 받는 학회가 되었다. 이렇듯 CVPR은 학계와 산업계 모두에서 크게 주목받으며 성장하는 학회이다.

본 학회에 마지막으로 참석했던 2013년 비교하여 올해 내가 받은 인상을 세가지만 뽑아서 정리해 보았다.

글 | 천영재 yeongjae.cheon@kakaobrain.com

지난 10년 동안 컴퓨터비전 분야(얼굴/사물 인식) 한 우물만 파왔다. 방법보다는 문제에 집중하며, 효율적이면서 효과적인 기술이다 싶으면 어느새 빠져드는 전형적인 엔지니어이다. 사람과 시간을 소중히 하며 끝까지 매진하여 좀 더 크고 가치 있는 결과물을 만들어내고 싶다.

모든 면에서 폭발적인 성장

먼저 워크숍이 열린 홀의 크기와 참석 인원에 놀라지 않을 수 없었다. 메인 학회에 앞선 워크숍 첫날 동시에 10여개의 워크숍들이 진행되었는데, 필자가 참석한 자율주행 관련 워크숍은 메인 학회에 조인트로 열리는 워크숍이라고 보기 어려울 정도로 상당히 큰 홀에서 진행되었다. 더군다나 워크숍 첫날은 보통 민망할 정도로 사람이 없기 마련인데 이번 워크숍에서는 첫날부터 발표장이 상당이 붐볐다. 메인 컨퍼런스는 3개의 세션으로 나뉘어 각자 다른 대형 홀에서 진행되었는데 등록자가 많다 보니 어느 세션에서도 빈자리를 찾기가 쉽지 않았다. 이러한 열기와 참여는 메인 학회와 워크숍 마지막 날까지 이어졌다.

한국에서도 많은 학생과 연구원들이 참석한 것을 보고 한번 더 놀랐다. 필자가 대학원 석사과정이었던 10년 전에는 논문을 발표하는 학생만 학회에 참석하는 것을 당연하게 생각했었는데, 이번 학회에는 한국의 복수의 대학원 연구실에서 논문이 없는 학생들에게도 학회에 참관할 수 있는 기회가 주어졌다. 또한 국책연구소나 회사에서도 많게는 십여 명의 연구원들이 오는 것을 볼 수 있었다. 덕분에 본인도 한국에서는 보기 힘들던 대학원 지도 교수님 및 선후배님들과의 모임을 가지기까지 하였다.

스폰서 기업과 이 기업들의 전시 부스도 이제는 학회 행사의 중요한 한 축을 담당하게 되었다. 수년 전 아기자기 하던 전시 부스들은 어디로 가고 이제는 상업 전시회를 방불케 할 정도로 큰 공간에서 다수의 글로벌 IT 기업들을 포함한 130여 기업의 부스가 차려졌다.

arXiv를 통한 출판 전 논문 공개와 오픈 소스 활성화

한두 해 전부터 이미 활성화 된 arXiv를 통한 논문의 출판 전 공개와 깃허브(github)를 통한 소스 공개 경향은 더욱 두드러져 보였다. 베스트 페이퍼를 받은 논문들은 말할 것도 없거니와 다수의 구두 발표 논문들이 이미 arXiv로 통해서 공개 된 것들이었다. 결과적으로 최신 논문들로 가득해야 할 학회 발표장에는 이미 많이들 읽어본 심지어는 피인용수가 50건을 넘긴 논문들을 찾아 볼 수 있었다. 이러한 이유에서인지 몇 해 전부터 본 학회는 발표하는 모든 논문을 온라인 상에 무료로 공개하고 있으며[*1], 작년부터는 튜토리얼과 구두발표, 스팟라이트 발표 동영상까지도 유튜브를 통해서 공개[*2]하고 있다.

저자가 소스를 깃허브(github)를 통해서 소스를 공개하는 모습도 쉽게 찾아 볼 수 있었다. 게다가 arXiv로 미리 공개된 유명 논문들은 저자가 소스코드를 공개하지 않더라도 독자들이 앞다투어 각자 다른 버전의 코드를 작성해 공개하고 있다. 한 예로, 최우수 논문상을 받은 DenseNet[*3]은 저자가 공개한 구현[*4] 외에도 15개 이상의 서로 다른 구현을 깃허브를 통해서 내려 받을 수 있다.

리크루팅(recruiting) 전쟁터

전시 부스의 가장 큰 목적은 리크루팅이라고 느껴졌다. 단순히 기술을 전시하고 홍보하는 것이 아니라, 적극적으로 학생들과 연구원들의 연락처를 수집하고 경우에 따라서는 현장에서 회사의 연구원과 학생 사이에 리쿠르팅 관련 질의 응답이 이루어지는 경우도 있었다.

저녁에는 우버, 구글, 트위터, 마이크로소프트 와 같이 주로 북미 회사들이 네트워킹 파티를 열었다. 나는 스냅챗(SnapChat) 파티에 초대 받아 참석하였는데 스냅챗 연구원들 뿐 아니라 실리콘 밸리의 벤처 창업자들이나 다른 회사 연구원들이 연구 주제와 현재의 소속 앞으로의 계획 등에 대하여 물어왔다. 게다가 일부 회사들은 비공개 소규모 리셉션을 한다거나 그룹 단위로 학생들을 초대하여 식사를 하는 경우도 많았던 것으로 알고 있다.

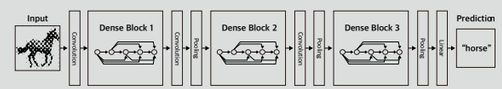
덕분에 학생들도 바쁜 일정을 소화해야 했을 것이다. 봐야할 포스터들도 많았을 뿐더러, 졸업을 앞둔 학생들은 여러 팀들과 인터뷰를 본다고, 또 저녁 시간에는 파티나 식사 모임에도 참석해야 했으니 말이다.

최우수 논문상을 받은 4편의 논문

1) 2편의 최우수 논문상(Best Paper Awards).

첫번째는 페이스북인공지능연구소(Facebook AI Research, FAIR)에서 작년 12월 발표한 '성기게 연결된 컨볼루셔널 네트워크(Densely Connected Convolutional Networks, DenseNet)'[*5]이었다. 본 논문은 ResNet[*6] 등과 같은 최신 컨볼루션 네트워크(convolutional networks) 연구에서 입력 레이어와 출력 레이어 사이에 짧은 연결들이 포함되면 더욱 깊은 구조의 모델을 효과적으로 학습하여 높은 정확도를 얻을 수 있다는 직관으로부터 시작한다. 논문에서 저자들은 더욱 성기게 연결되는(densely connected) 단순한 구조의 모델을 제안하고 있다. 결과적으로 DenseNet은 물체 인식(object recognition) 벤치마크 태스크에서 발표 당시 빠르면서도 가장 좋은 성능을 보였다.

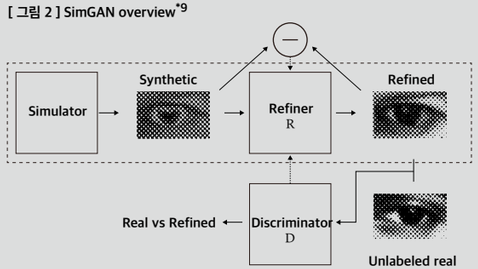
[그림 1] Densely connected convolutional network[*7]



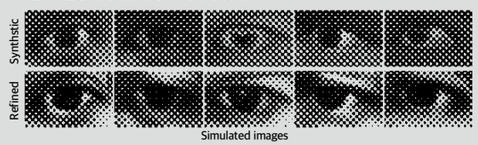
두번째는 작년 12월 애플이 비밀주의를 깨고 최초로 논문을 공개하여 화제가 되었던 '적대적 훈련을 통해 시뮬레이션된 이미지와 감독되지 않은 이미지로부터의 학습(Learning from Simulated and Unsupervised Images through Adversarial Training)'[*8]이다. 지도 학습(supervised learning)은 불가피하게 대량의 학습 이미지와 함께 정답(ground truth)을 필요로 하는데 이는 많은 경우 시간과 돈의 문제로 제한을 받는다. 이를 해결하기 위한 방법 중 최근 가장 각광을 받고 있는 방법이 본 논문에서와 같이 실제 데이터가 아닌 그래픽스 엔진을 이용하여 데이터를 합성하여 사용하는 방법이다.

하지만 이러한 방법은 필연적으로 실제 데이터와 합성된 데이터 사이의 차이로 인한 성능 저하 문제를 가지게 된다. 본 논문은 [그림 2]에서와 같이 합성된 이미지를 실제 이미지들과 구분하기 어려우면서도 합성이미지와도 유사한 이미지로 수정하는 Refiner를 적대적 훈련(adversarial training) 방법으로 학습함으로 이 문제를 해결하고 있다. [그림3] 에서 합성이미지가 보다 사실적으로(realizm) 개선되는 것을 확인 할 수 있다. 이 방법을 시선 추정(gaze estimation) 문제와 손동작 추정(hand pose estimation)문제에 적용하여 기존 방법 대비 성능 개선을 얻을 수 있었다.

[그림 2] SimGAN overview[*9]



[그림 3] 합성이미지와 수정된 이미지 예시[*10]



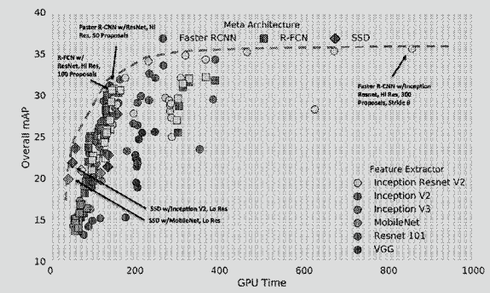
2) 2편의 최우수 영애 멘션상(Best Paper Honorable Mention Awards)

일반적으로 학회 조직 위원회(committee)에 의하여 선정되는 최우수 영애 멘션상은 Polygon-RNN을 이용하여 미지에서 물체에 주석(annotation)을 만드는 작업을 적은 수의 수작업으로 가능하게 하는 방법을 제안한 '다각형-RNN을 이용한 물체에 주석달기(Annotating Object Instances with a Polygon-RNN)'[*11]와 모델의 크기를 조절하여 속도와 성능을 상대적으로 다양하게 조정할 수 있으면서도 어떠한 경우에도 기존의 다른 알고리듬보다 빠르면서 좋은 성능을 보장하는 '요로9000(YOLO9000: Better, faster, stronger)'[*12]이 수상하였다.

3) 그밖에 인상 깊었던 논문들

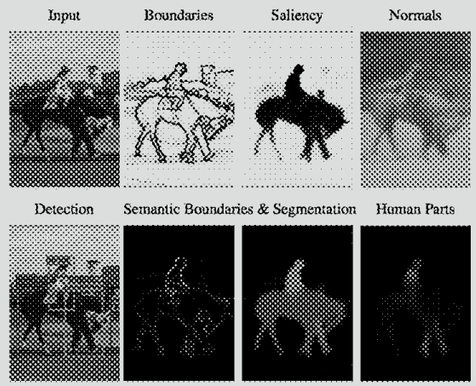
(1) '최신 컨볼루션 물체 검출기들의 속도와 정확도 사이의 트레이드오프(Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors)'[*13] (Google) : 이미지에서 물체를 검출하는 문제를 풀고자 할 때, 주어진 어플리케이션이나 플랫폼에 따라서 속도, 메모리, 성능 등의 제약 조건이 다를 수 있다. 이때 널리 알려진 다양한 딥러닝 모델 구조 중 옳은 선택을 하기 위한 가이드를 제공하고 있다. 뿐만 아니라 제안하고 있는 딥러닝 모델구조와 하이퍼파라미터를 쉽게 조정할 수 있는 구현 코드 모두를 공개하고 있다.

[그림 4] COCO dataset에서 150개 물체 검출기의 속도와 성능 비교 그래프[*14]



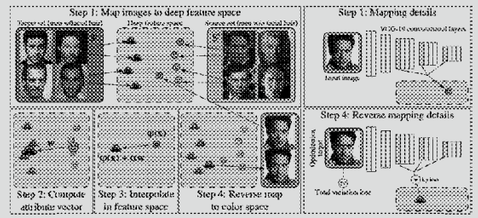
(2) '우버넷(UberNet: Training a Universal Convolutional Neural Network for Low-, Mid-, and High-Level Vision Using Diverse Datasets and Limited Memory)'[*15] (FAIR) : 마치 스위스칼(swiss knife)과 같이 컴퓨터 비전 분야에서의 '현저한 물체 추정(saliency estimation)', '외각선 검출(boundary detection)', '물체 검출(object detection)', '시맨틱 분할(semantic segmentation)', '물체 파트 검출(object part detection)' 등의 다양한 레벨의 문제를 하나의 모델로 풀 수 있도록 학습하는 방법을 제안하고 있다.

[그림 5] Low-, Mid-, and High-Level Vision tasks[*16]



(3) '이미지 콘텐츠 변화를 위한 깊은 특징점 보간법(Deep Feature Interpolation for Image Content Changes)'[*17] : [그림6] 에서와 같이 깊은 컨볼루셔널 피쳐(deep convolutional feature) 공간에서 보간법(Interpolation)을 이용하여 이미지 컨텐츠(나이 혹은 표정)를 변경하는 방법을 제안하고 있다. [그림7]은 입력 얼굴을 나이든 얼굴로 변형한 예시이다.

[그림 6] 'Deep feature interpolation' 구조도[*18]

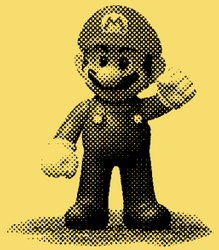


[그림7] Ageing 예시[*19]

글을 마치며

arXiv 덕분에 유명 논문들은 학회 전에 이미 읽어 볼 수 있고 나머지 논문들과 심지어 발표 동영상까지 올라오는데 왜 굳이 많은 비용과 시간을 들여서 학회에 가는지를 누군가 물을 수 있을 것이다. 하지만 '백문이 불여일견'이라는 속담처럼 가서 보고 느끼고 네트워킹하며 받는 자극은 분명 단순히 논문을 읽는 것과는 비교할 수 없는 가치가 있을 것이다. 본인은 앞으로도 머신러닝(특히 딥러닝)과 컴퓨터 비전 모두를 동시에 다루는 가장 큰 학회인 CVPR은 꼭 참석하고 싶다. 이 자리를 빌려 학회를 참관할 수 있는 기회를 준 회사와 동료들에게 감사의 마음을 전한다.

***1** 참고 | http://openaccess.thecvf.com/CVPR2017.py ***2** 참고 | https://www.youtube.com/channel/UC0n76gicaarsN_Y9YShWwhw/videos ***3** 논문 | Huang, G., Liu, Z., Weinberger, K. & Maaten, L. (2017). Densely connected convolutional networks. CVPR. ***4** 참고 | https://github.com/liuzhuang13/DenseNet ***5** 논문 | Huang, G., Liu, Z., Weinberger, K. & Maaten, L. (2017). Densely connected convolutional networks, CVPR. ***6** 논문 | He, K., Zhang, X., Ren S. & Sun, J. (2016). Deep Residual Learning for Image Recognition, ECCV. ***7** 논문 | Huang, G., Liu, Z., Weinberger, K. & Maaten, L. (2017). Densely connected convolutional networks (p. 3), CVPR. ***8** 논문 | Shrivastava, A. et al. (2017). Learning from simulated and unsupervised images through adversarial training, CVPR. ***9** 논문 | Shrivastava, A. et al. (2017). Learning from simulated and unsupervised images through adversarial training (p. 2), CVPR. ***10** 논문 | Shrivastava, A. et al. (2017). Learning from simulated and unsupervised images through adversarial training (p. 5), CVPR. ***11** 논문 | Castrejon, L., Kundu, K., Urtasun, R. & Fidler, S. (2017). Annotating object instances with a Polygon-RNN, CVPR. ***12** 논문 | Redmon, J. & Farhadi, A. (2017). YOLO9000: Better, faster, stronger, CVPR. ***13** 논문 | Huang, J. et al. (2017). Speed/Accuracy trade-offs for modern convolutional object detectors, CVPR. ***14** 논문 | Huang, J. et al. (2017). Speed/Accuracy trade-offs for modern convolutional object detectors (p. 9), CVPR. ***15** 논문 | Kokkinos, I. (2017). UberNet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. CVPR. ***16** 논문 | Iasonas Kokkinos. UberNet: Training a Universal Convolutional Neural Network for Low-, Mid-, and High-Level Vision Using Diverse Datasets and Limited Memory. In CVPR, 2017. ***17** 논문 | Upchurch, P. et al. (2017). Deep Feature Interpolation for Image Content Changes, CVPR. ***18** 논문 | Upchurch, P. et al. (2017). Deep Feature Interpolation for Image Content Changes (p.2), CVPR. ***19** 논문 | Upchurch, P. et al. (2017). Deep Feature Interpolation for Image Content Changes (p.1), CVPR.



슈퍼마리오 그리고 GAN

AI 학습 체계에서 강화학습, 그리고 GAN은 현재 가장 주목받는 개념이라고 해도 과언이 아닐 겁니다. 알파고의 학습 체계로 유명해진 강화학습. 카카오AI리포트는 강화학습이 무엇인지에 대해 앞서 설명드린 바 있기도 합니다. 이번에는 강화학습을 전설의 게임인 슈퍼마리오에 적용한 후기를 소개합니다. 총 3회에 걸쳐 연재될 송호연 님의 슈퍼마리오와 강화학습 시리즈는 국내 AI커뮤니티에서 올 하반기 두고두고 회자될 것으로 예상합니다. 그리고, 강화학습에 이어 실증적 차원에서 GAN의 개념을 소개한 유재준 님의 글이 2회에 걸쳐 소개됩니다. GAN의 개념, 그리고 기본적인 수학 개념을 중심으로 기술하신 유재준 님의 목표는 "추후 새로운 GAN 연구를 봤을 때 관련 흐름을 따라갈 수 있는 기초적인 지식을 제공하는 것"이었습니다.

강화학습으로 풀어보는 슈퍼마리오 part 1.

강화학습(reinforcement learning)은 머신러닝의 꽃이라고도 불린다. 강화학습의 매력에 깊이 빠진 필자는 더 많은 분들에게 강화학습의 '복음'을 전하기 위해 튜토리얼을 작성하고자 한다. 지금 IT업계에서 일하고 계시는 분들 중 많은 분이 경험하셨을법한 유명한 게임, '슈퍼마리오'를 강화학습 튜토리얼의 테마로 잡았다. 딥마인드(Deepmind)가 공개한 DQN(Deep Q Network) 알고리듬으로부터 최신 트렌드인 A3C까지 여러 가지 알고리듬을 하나씩 소개하면서 슈퍼마리오라는 게임 세상을 강화학습으로 풀어 보도록 하자. 슈퍼마리오에 강화학습 적용하기는 총 세 번에 걸쳐 연재된다.

튜토리얼을 통해 여러분은 이론이 아닌 현실에서 돌아가는 프로그램을 볼 수 있게 될 것이다. 우선 트레이닝이 되는 프로그램을 따라서 돌려 본 후에 이론을 설명하는 방식으로 진행해서, 조금 더 쉽고 흥미롭게 강화학습을 공부하실 수 있도록 진행할 예정이다.

글 | 송호연 chris.song@kakaocorp.com

Kakao R&D Center Data Engineer.
강화학습과 썸을 타다가 최근에 연애를 시작했습니다. 자기 전에도, 아침에 일어났을 때도 강화학습 생각이 납니다. 앞으로 세상을 파괴적으로 혁신시킬 범용인공지능(Artificial General Intelligence)에 완전히 빠져있습니다. 카카오에서는 대규모 데이터 유저 프로파일링, 머신러닝 기술을 활용해 현실의 문제를 해결하는 데이터 엔지니어로 일하고 있습니다.

가장 먼저, 강화학습의 환경을 먼저 세팅해 보도록 하자.

준비물
- Anaconda3 (Python3.6)
- Homebrew (MacOS 유저)
- GIT
- pip

Pip 라이브러리
- https://github.com/chris-chris/gym-super-mario (forked from ppaquette/gym-super-mario)
- https://github.com/openai/baselines
- https://github.com/openai/gym

게임 에뮬레이터
- FCEUX (http://www.fceux.com)

깃허브(Github) 프로젝트
- https://github.com/chris-chris/mario-rl-tutorial

슈퍼마리오 강화학습 실행 환경 설정

강화학습 스크립트가 실행이 가능하도록 환경설정을 성공하는 것이 첫 번째 목표다. 게임 실행이 되어야 강화학습 코드도 작성하고 하이퍼파라미터도 튜닝할 수 있을 테니 말이다. 오픈AI(OpenAI) 짐(gym)[*1]에 설치된 기본 아타리(Atari) 게임 환경들이 있지만, 좀 더 재미있는 슈퍼마리오 강화학습 환경을 만들기 위해선 조금 더 수고로운 환경설정이 필요하다. 이 튜토리얼은 맥OS(MacOS) 환경 기준으로 진행한다(윈도우의 경우엔 cygwin과 fceux의 조합을 활용해서 설정이 가능하다). (※지면에 다 담기 어려운 점을 고려해 설치가 쉬운 아나콘다3(Anaconda3) 설치와 기본적인 GIT 사용법은 설명하지 않겠다.)

가장 먼저 fceux 를 설치해야 한다. fceux는 PC 환경에서 오래된 고전 게임들을 실행할 수 있는 에뮬레이터다. 가장 간단하게 설치하는 방법은 바로 홈브류(Homebrew)를 활용한 설치법이다. 홈브류가 깔려 있지 않다면, 터미널(Terminal)을 열고 간단하게 아래 명령어를 실행해서 설치가 가능하다. (홈브류의 설치 가이드는 https://brew.sh 에서 확인할 수 있다)

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/
Homebrew/install/master/install)"
```

홈브류 설치가 완료되면, 다시 터미널을 열고 아래 명령어로 fceux를 설치하자.

```
brew install fceux
```

위 명령어를 실행하면 다음과 같은 결과가 표시된다. 홈브류를 활용하면 다양한 패키지를 편하게 설치할 수 있다.

[그림 1] fceux 설치

```
:Chrisui-MacBook-Pro:~ Chriss brew install fceux                                :
Updating Homebrew...
==> Downloading https://homebrew.bintray.com/bottles-portable/portable-ruby-2.3.
3.leopard_64.bottle.1.tar.gz
######################################################################## 100.0%
==> Pouring portable-ruby-2.3.3.leopard_64.bottle.1.tar.gz
==> Auto-updated Homebrew!
Updated 1 tap (homebrew/core).
==> New Formulae
abyss          beast          bowtie2        fn             xmrig          zim
```

Fceux 프로그램을 설치했다면, 터미널을 열고 아래 명령어들을 실행해서 필요한 pip 라이브러리들을 설치한다.

첫 번째로 필요한 pip 패키지는 OpenAI의 gym이다.

```
pip install gym
```

그리고 이미지 처리를 위한 opencv-python 패키지를 설치한다.

```
pip install opencv-python
```

그다음 설치할 pip 패키지는 OpenAI의 baselines[*2]다. 주의할 점은, pip install baselines 명령으로 baselines 를 설치하면 최신 버전을 받을 수 없다는 것이다. 깃허브에 올라와 있는 최신 빌드를 설치하기 위해선 아래 명령으로 설치하기 바란다.

```
pip install git+https://github.com/openai/baselines
```

그리고 설치할 pip 패키지는 슈퍼마리오의 다양한 난이도가 들어 있는 gym-super-mario 패키지다. 원래 이 환경은 필립 파퀘트(Phillip Paquette, @ppaquette)[*3]가 제작했으며, 파퀘트의 gym-super-mario에서 멀티 에이전트(multi agent) 처리 등 몇 가지 이슈를 수정하기 위해 포크(fork)한 후 코드를 수정했다. 아래 명령어를 실행해서 필자가 수정한 코드로 환경을 설치하자.

```
pip install git+https://github.com/chris-chris/gym-super-mario
```

그리고 슈퍼마리오 강화학습 예제 프로젝트를 클론(clone)하자.

```
git clone https://github.com/chris-chris/mario-rl-tutorial
```

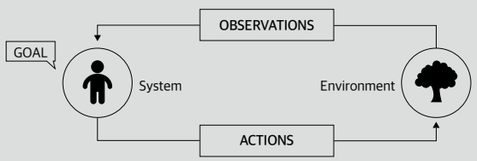
이제 클론을 받은 프로젝트 폴더로 들어간 후 아래 명령어를 실행하면 슈퍼마리오 에뮬레이터가 화면에 뜨면서 학습이 시작된다.

```
python train.py
```

강화학습 Observation Space 설정

이제 슈퍼마리오의 학습 환경에 대해 설명한다. 딥마인드에서 발표한 슬라이드에서 잘 나타나 있듯이, 강화학습의 학습환경은 아래와 같이 Observations, Actions 두 가지 정보로 이루어진다.

[그림 2] 강화학습 프레임워크[*4]



이제 observation space와 action space 각각이 어떻게 처리되는지를 알아보자.

첫 번째로, observation space는 다른 Atari 게임들과 마찬가지로 화면의 RGB픽셀을 그대로 입력 받는다. 슈퍼마리오의 observation space의 구체적인 설정은 아래와 같다.

```
spaces.Box(low=0, high=255, shape=(224, 256, 3))
```

화면의 세로 사이즈(height)는 224, 가로 사이즈(width)는 256, 색상의 종류(RGB)는 3, 그리고 각 색상의 수치 값의 범위는 0부터 255까지다. 실제로, 화면의 픽셀데이터를 받아서 출력해 보면 아래와 같이 나온다.

```
[[[200  76  12]
  [200  76  12]
  [200  76  12]
  ...,
  [252 188 176]
  [200  76  12]
  [200  76  12]]]
```

하나의 픽셀에 RGB 값 3가지가 들어 있다. 224×255 2차원 행렬 안에 각 픽셀이 3가지 RGB 값을 가지고 있으니 3차원 행렬이 된다.

강화학습에서는 이렇게 들어오는 데이터를 CNN(Convolution Neural Network)에 넣어서 강화학습 모델이 화면의 여러 피처들을 인식할 수 있도록 한다. 그런데 observation space데이터를 CNN에 넣기 전에 중요한 과정이 있다. 바로, 데이터의 사이즈를 줄이는 것이다. OpenAI gym상에서 구동되는 아타리(Atari) 게임 환경들과 예시들을 보면 RGB를 그레이스케일로 변환하고 이미지 사이즈를 줄이는 코드를 확인할 수 있다.

일반적으로 이렇게 RGB값이 들어오면, 연산을 단순화하기 위해 RGB값을 그레이스케일(회색)로 변환한다. 그리고 이미지 사이즈를 84 × 84 사이즈로 줄인다.

```
wrapper.py : 화면에 출력된 RGB 이미지 데이터를 그레이스케일로 변환하고 크기를 줄이는 Wrapper

import cv2
import gym
import numpy as np
from gym import spaces

class ProcessFrame84(gym.ObservationWrapper):
 def __init__(self, env=None):
  super(ProcessFrame84, self).__init__(env)
  self.observation_space = spaces.Box(low=0, high=255, shape=(84, 84, 1))

 def _observation(self, obs):
  return ProcessFrame84.process(obs)

 @staticmethod
 def process(img):
  img = img[:, :, 0] * 0.299 + img[:, :, 1] * 0.587 + img[:, :, 2] * 0.114
  x_t = cv2.resize(img, (84, 84), interpolation=cv2.INTER_AREA)
  x_t = np.reshape(x_t, (84, 84, 1))
  x_t = np.nan_to_num(x_t)
  return x_t.astype(np.uint8)
```

이렇게 화면의 RGB값을 전처리하여 작은 데이터 사이즈로 만들어 보면, 아까 예시로 들었던 행렬 데이터에서 RGB데이터는 그레이스케일로 변하면서 3배수로 줄어들었고(1/3로 감소), 공간 사이즈(가로×세로)는 약 8.12배수로 줄어들었다(약 12.3% 감소). 이렇게 피처를 인식할 데이터의 크기를 상당히 줄여서 우리는 강화학습 모델을 좀 더 효율적으로 학습시킬 수 있게 된다.

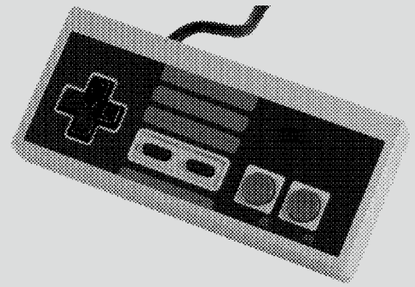
강화학습 Action Space 설정

이제 action space를 살펴보자. Action space는 강화학습 모델이 에이전트에 명령을 전달할 때 명령의 가짓수다. Atari 게임 중 Pong의 경우엔 action space가 정말로 작다. 위로 가거나 아래로 가면 된다. 하지만, 슈퍼마리오는 그보다 좀 더 복잡한 action space를 가지고 있다. 바로 방향키와 A, B 버튼이다.

· 방향키: 상, 하, 좌, 우 4가지
· 버튼: 버튼은 A, B 2가지

그렇다면 action space의 크기는 6일까? 아니다. 우선, 슈퍼마리오에서 우리가 흔히 내리는 명령들을 한번 생각해 보자. 우리는 오른쪽 방향키를 눌러서 앞으로 가면서 동시에 A 버튼을 눌러서 점프를 할 수 있다. 우리는 6가지 명령을 동시에 내릴 수 있다. 그러면 갑자기 space가 엄청나게 커진다. 2의 6제곱이 된다. 최대 경우의 수는 64다. 하지만, 64가지의 경우의 수가 모두 쓸모 있지는 않다. 예를 들어, 위 방향키와 아래 방향키를 굳이 같이 누를 필요가 없으니 말이다.

[그림 3] 게임용 콘솔(console)



흔히 쓰일 만한 action space를 14가지로 추려 보았다. 14가지 각각의 명령 조합은 아래와 같다.

```
0 : 아무것도 안 함
1 : 위
2 : 아래
3 : 왼쪽
4 : 왼쪽 + A
5 : 왼쪽 + B
6 : 왼쪽 + A + B
7 : 오른쪽
8 : 오른쪽 + A
9 : 오른쪽 + B
10 :  오른쪽 + A + B
11 : A
12 : B
13 : A + B
```

이렇게 14개의 조합을 만들어 놓고 게임 환경에 명령을 보낼 때는 각각의 명령을 동시에 입력 가능한 형식으로 변환시켜서 전달하게 만들 것이다.

예를 들어 오른쪽 키를 누르는 명령과 A 키를 동시에 누르는 8번 명령은 [0, 0, 0, 1, 1, 0] 이런 형식의 명령으로 변환되어 게임 에이전트에게 전달된다. 그리고 왼쪽으로 움직이는 3번 명령은 [0, 1, 0, 0, 0, 0] 데이터로 변환되어 에이전트에 전달된다. 이렇게 14가지 명령을 게임 에이전트가 알아듣기 좋게 변환시키는 래퍼(Wrapper)가 바로 MarioActionSpaceWrapper다.

```
MarioActionSpaceWrapper : 슈퍼마리오 게임의 Action Space를 재정의해주는 gym Wrapper

class MarioActionSpaceWrapper(gym.ActionWrapper):

 mapping = {
  0: [0, 0, 0, 0, 0, 0],  # NOOP
  1: [1, 0, 0, 0, 0, 0],  # Up
  2: [0, 0, 1, 0, 0, 0],  # Down
  3: [0, 1, 0, 0, 0, 0],  # Left
  4: [0, 1, 0, 0, 1, 0],  # Left + A
  5: [0, 1, 0, 0, 0, 1],  # Left + B
  6: [0, 1, 0, 0, 1, 1],  # Left + A + B
  7: [0, 0, 0, 1, 0, 0],  # Right
  8: [0, 0, 0, 1, 1, 0],  # Right + A
  9: [0, 0, 0, 1, 0, 1],  # Right + B
  10: [0, 0, 0, 1, 1, 1],  # Right + A + B
  11: [0, 0, 0, 0, 1, 0],  # A
  12: [0, 0, 0, 0, 0, 1],  # B
  13: [0, 0, 0, 0, 1, 1],  # A + B
 }

 def __init__(self, env):
  super(MarioActionSpaceWrapper, self).__init__(env)
  self.action_space = spaces.Discrete(14)

 def _action(self, action):
  return self.mapping.get(action)

 def _reverse_action(self, action):
  for k in self.mapping.keys():
   if(self.mapping[k] == action):
    return self.mapping[k]
  return 0
```

우리가 만들어 본 observation space wrapper와 action space wrapper를 환경에 적용하는 법은 간단하다. 아래 코드와 같이 실행하면 된다.

```
- train_dqn

환경을 생성하고
Action Space Wrapper를 적용하고
Observation Space Wrapper를 적용한 코드

def train_dqn(env_id, num_timesteps):
 """Train a dqn model.

  Parameters
  -------
  env_id: environment to train on
  num_timesteps: int
      number of env steps to optimizer for

 """

 # 1. Create gym environment
 env = gym.make(FLAGS.env)
 # 2. Apply action space wrapper
 env = MarioActionSpaceWrapper(env)
 # 3. Apply observation space wrapper to reduce input size
 env = ProcessFrame84(env)
```

MarioActionSpaceWrapper는 action space를 커스터마이징하는 wrapper이고, ProcessFrame84는 observation space의 사이즈를 줄이는 wrapper이다. 이제 학습을 위한 action space와 observation space 설정을 완료했다.

강화학습 Q-Function 모델 구조 설정

이제 Q-Function 모델에 대해서 살펴보자.

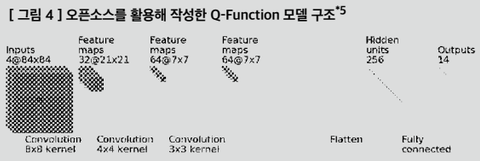
```
- train_dqn

환경을 생성하고
Action Space Wrapper를 적용하고
Observation Space Wrapper를 적용한 후
```

CNN 모델을 적용한 Q 함수를 생성한 코드

```
def train_dqn(env_id, num_timesteps):
 """Train a dqn model.

  Parameters
  -------
  env_id: environment to train on
  num_timesteps: int
     number of env steps to optimizer for

  """

  # 1. Create gym environment
  env = gym.make(FLAGS.env)
  # 2. Apply action space wrapper
  env = MarioActionSpaceWrapper(env)
  # 3. Apply observation space wrapper to reduce input size
  env = ProcessFrame84(env)
  # 4. Create a CNN model for Q-Function
  model = cnn_to_mlp(
  convs=[(32, 8, 4), (64, 4, 2), (64, 3, 1)],
  hiddens=[256],
  dueling=FLAGS.dueling
  )
```

우리는 총 3개의 convolution layer를 사용할 것이다. 그리고 마지막에 CNN Layer를 Flatten 방식으로 1자로 펼친 후 fully connected 방식으로 14개의 action space에 연결할 것이다. 그림으로 그려서 설명하는 게 가장 이해하기 쉬울 것 같아 오픈소스로 작성된 시각화 예제를 활용하여 Q-function을 그려 보았다.

[그림 4] 오픈소스를 활용해 작성한 Q-Function 모델 구조*5



우리는 3개의 CNN 레이어를 활용해 화면상의 물체를 식별한 후, 식별한 결과를 256개 유닛으로 이루어진 fully connected로 연결한 후 최종적으로 14개의 결과를 받아 오도록 하였다.

여기서 84×84 흑백 이미지가 4겹으로 된 것을 확인할 수 있는데, 바로 최근 4프레임의 화면을 쌓아서 입력값으로 넣은 것이다. 4개의 84×84 사이즈 흑백 이미지를 14개의 명령 신호와 연결시키는 모델을 완성한 것이다.

랜덤 에이전트(Random Agent) 만들기

실제로 움직이는 에이전트(agent)를 만들어서 돌려 보자. 랜덤 에이전트는 각 단계(step)마다 취할 수 있는 모든 행동(action)을 랜덤으로 골라서 명령을 보내는 에이전트이다. 이것을 강화학습이라 할 수는 없지만 게임 환경(enviroment)에서 action Space에 대한 프로세스를 이해하기 위해 돌려 보도록 한다.

random_agent.py : 랜덤으로 명령을 보내는 agent 예제

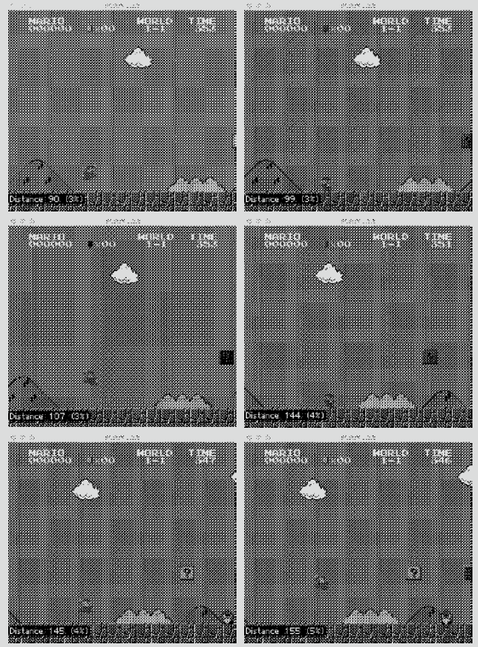
```
import gflags as flags
import sys
import gym

import ppaquette_gym_super_mario

from wrappers import MarioActionSpaceWrapper, ProcessFrame84

FLAGS = flags.FLAGS
flags.DEFINE_string("env", "ppaquette/SuperMarioBros-1-1-v0", "RL
environment to train.")

class RandomAgent(object):
 """The world's simplest agent!"""
 def __init__(self, action_space):
   self.action_space = action_space

 def act(self, observation, reward, done):
   return self.action_space.sample()

def main():
 FLAGS(sys.argv)
 # Choose which RL algorithm to train.

 print("env : %s" % FLAGS.env)

 # 1. Create gym environment
 env = gym.make(FLAGS.env)
 # 2. Apply action space wrapper
 env = MarioActionSpaceWrapper(env)
 # 3. Apply observation space wrapper to reduce input size
 env = ProcessFrame84(env)

 agent = RandomAgent(env.action_space)

 episode_count = 100
 reward = 0
 done = False

 for i in range(episode_count):
   ob = env.reset()
   while True:
     action = agent.act(ob, reward, done)
     ob, reward, done, _ = env.step(action)
     if done:
       break

if __name__ == '__main__':
 main()
```

랜덤 에이전트의 알고리듬은 단순하다. 매 스텝마다 슈퍼마리오 게임 환경에서 취할 수 있는 14개의 action space 중 하나를 랜덤으로 골라서 명령을 보내는 것이다. 이 random_agent.py를 실행하기 위해서는 아래 명령어를 실행하면 된다.

```
python random_agent.py
```

그러면, 화면*6에서 랜덤으로 움직이는 슈퍼마리오를 확인할 수 있다.

[그림 5] 랜덤으로 움직이는 슈퍼마리오*7



Agent 학습시키기

학습을 시킬 땐 간단히 아래 명령어를 실행하면 된다. 프로그램이 실행되면 화면에 슈퍼마리오가 실행되면서 학습이 진행된다.

```
python train.py --log=stdout
```

위 명령으로 실행하면, 학습 결과가 콘솔에 찍힌다. 하지만 기록이 길어지다 보면 보기가 불편할 것이다. 그래서 텐서보드(tensorboard)로 확인할 수 있는 방법을 소개한다.

```
python train.py --log=tensorboard
```

위 명령어를 실행시키면, 프로젝트 폴더 내에 텐서보드 폴더가 생성될 것이다. 여기에는 텐서보드에 맞는 로그(log)가 쌓이기 시작한다. 이 학습로그를 텐서보드를 띄워서 확인해 보자.

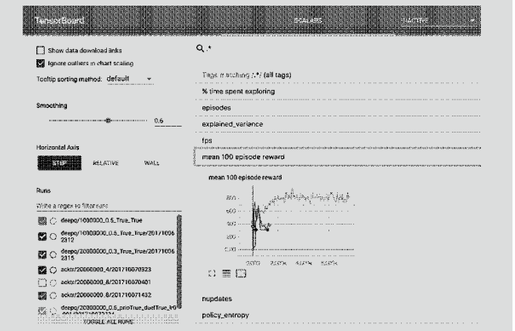
```
tensorboard logdir=tensorboard
```

이렇게 명령어를 실행하면, 아래와 같이 콘솔에 찍힐 것이다.

```
$tensorboard --logdir=tensorboard
TensorBoard 0.1.6 at http://Chrisui-MacBook-Pro.local:6006 (Press CTRL+C
to quit)
```

그러면 브라우저를 열고 이렇게 이 주소(http://localhost:6006)로 이동해 보자. 그러면 텐서보드에서 학습결과를 확인할 수 있을 것이다.

[그림 6] 텐서보드를 통해 확인한 학습 결과



학습된 Agent 실행시키기

학습을 진행하면서 최근 100건의 평균 보상(reward)이 최고치를 경신하면, 현재 학습된 모델을 파일로 저장하도록 만들어 두었다. 프로젝트 폴더 내에 models/deepq/mario_reward_930.6.pkl 파일이 들어 있는데, 이 모델로 실행시켜 보도록 하자.

```
python enjoy.py --algorithm=deepq --file=mario_reward_930.6.pkl
```

위 명령을 실행하면 학습된 모델을 실행시켜 볼 수 있다. --file= 파라미터에 본인이 만들어 낸 파일명 매개변수로 입력하면 된다.

이렇게, 슈퍼마리오 게임에 강화학습 에이전트를 구현하기 위한 세부 작업을 설명하였다. 다음 편에서는 DQN 강화학습의 이론에 대해서 설명한 후, Prioritized Replay Memory, Dueling DQN 등 점진적으로 개선된 방법론을 이론과 같이 예제를 하나씩 소개하고자 한다.

*1 참고 | https://github.com/openai/gym *2 참고 | https://github.com/openai/baselines *3 참고 | https://github.com/ppaquette/gym-super-mario *4 참고 | Demis Hassabis, CEO, DeepMind Technologies - The Theory of Everything, Youtube, https://www.youtube.com/watch?v=rbsqaJwpu6A *5 참고 | https://github.com/gwding/draw_convnet *6 편집자 주 | [그림 5]라고 표기했지만, 실제 필자가 집필진에서 넘겨 주신 파일은 움직이는 영상입니다. 물리적 한계 상 필자가 의도하는 바를 간행물에는 온전히 담지 못했습니다. 해당 영상은 온라인에는 온전히 담았습니다. 다음 링크를 참고해 주세요(https://brunch.co.kr/@kakao-it/124/). *7 참고 | https://youtu.be/bicOms7rkcA

Do you know GAN? (1/2)

최근 딥러닝 분야에서 가장 뜨겁게 연구되고 있는 주제인 GAN(Generative Adversarial Network)을 소개하고 학습할 수 있는 글을 연재하려고 한다. GAN은 이름 그대로 뉴럴 네트워크(neural network)를 이용한 생성 모델이다. 흔하게는 이미지를 생성하는 것에서부터 음성, 문자에 이르기까지 다양한 분야에 적용되고 있다. 2014년에 이안 굿펠로우(Ian Goodfellow)가 GAN을 처음 선보였을 때부터 이미 대박의 '낌새'가 보였지만, NIPS 2016 이후부터는 GAN에 대한 관심이 더욱 폭발적으로 늘어나고 있다. 원래도 핫한 기계학습 분야에서도 GAN에 대한 연구는 유독 빠르게 발전하고 있다. 일주일이 멀다 하고 새로운 아이디어를 사용한 논문이 나오는 한편, 재미있는 방향의 관련 어플리케이션들이 속속들이 나오면서 GAN의 기세는 더해 가고 있다. 페이스북의 얀 레쿤(Yann Lecun) 교수도 GAN을 최근 10~20년 간에 기계학습 분야에서 나온 아이디어 중 최고라는 찬사를 보내는 만큼, 그 열기가 쉽게 식지는 않을 것 같다.

이번 호에서는 이렇게 사람들이 열광하는 GAN이 무엇인지, 어떤 원리로 그림을 생성하는지에 대해 먼저 소개하고자 한다. 그리고 독자들이 GAN의 개념과 그 밑에 깔려 있는 기본적인 수학적 배경을 이해하게 하고, 추후 새로운 GAN 연구를 봤을 때 관련 흐름을 따라갈 수 있는 기초적인 지식을 제공하는 것을 목표로 합니다.

글 | 유재준 jaejun2004@gmail.com

카이스트 바이오및뇌공학과 학부를 졸업하고 동 대학원 박사 과정을 밟고 있는 공돌이. 바이오영상신호처리 연구실에서 전통적인 신호처리 이론을 이용한 영상 복원이나 대수적 위상수학을 이용한 뇌 네트워크 분석을 연구하였으나, 최근에는 기계 학습에 빠져서 이를 이용한 의료 영상 복원에 적용해보고 있다. 처음 가본 NIPS 2016 학회에서 GAN을 접하고 매우 흥분하여 열심히 가지고 놀아보고 있지만 사실 당장 졸업에 필요한 연구 주제와는 전혀 관계가 없다는 것이 함정. 딥러닝은 이제 갓 1년차 공부 중인 초짜 대학원생이며, 혼자 공부하던 것을 블로그에 정리하던 것이 취미였는데 많은 분들이 좋아해 주셔서 매우 신이 나있다. 최근 졸업이 다가오면서 외부활동이 점차 줄어들고 있지만, 앞으로 더 많은 분들과 교류하고 배워 이 분야에서 내가 아는 것을 꾸준히 나눌 수 있는 사람이 되는 것이 꿈이다.

Generative Adversarial Networks

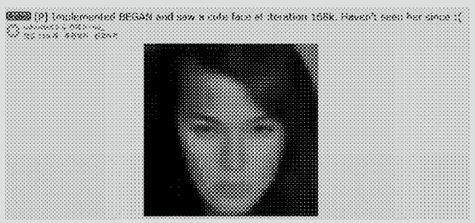
GAN 이전에도 생성 모델(generative model)은 매우 다양하게 있었는데 왜 하필 GAN만 유독 이렇게 난리일까? "잘되니까."

[그림 1]은 구글에서 올해 3월에 발표한 BEGAN 모델로 생성한 이미지들이다.

[그림 1] BEGAN 결과

즉, 이 사진들은 모두 실존하는 인물의 것이 아니란 것이죠. 상당히 그럴듯하지 않은가? 자세히 뜯어봐도 실제 사람의 사진인 것으로 착각할 정도이다. 워낙 (이미지 생성이) 잘되다 보니 우리나라로 치면 디시인사이드(dcinside.com)와 같은 레딧(Reddit)의 'Machine Learning' 게시판에서는 한때 [그림2]와 아래의 글이 게시되어 세간의 눈길을 끌기도 했다.

[그림 2] Reddit에 게시된 이미지



"BEGAN으로 학습시키던 중 16만8천 번째쯤 귀여운 그녀를 보았습니다. 그리고... 다시는 보지 못했어요. :("

사실 GAN이란 모델을 이렇게 잘 학습시키는 것은 쉬운 일이 아니다. 그 이유에 대해서는 나중에 차차 설명하도록 하고, GAN의 원리에 대해 먼저 소개하고자 한다.

GAN의 개념

GAN은 이름만 뜯어봐도 큰 줄기를 알 수 있다:

· Generative : CNN을 이용한 이미지 분류기(classifier)와 같이 이미지의 종류를 구별하는 것이 아닌 이미지를 만들어 내는 방법을 배우는 '모델'이라는 것을 알 수 있다.

· Adversarial: 이 단어의 사전적 의미는 "대립하는, 적대하는"이다. 대립하려면 상대가 있어야하니 GAN은 크게 두 부분으로 나뉘어 있다는 것을 직관적으로 알 수 있다.

· Network : 뉴럴 네트워크을 사용해서 모델이 구성되어 있다.

GAN은 이미지를 만들어 내는 네트워크(generator)와 이렇게 만들어진 이미지를 평가하는 네트워크(discriminator)가 있어서 서로 대립(adversarial)하며 서로의 성능을 점차 개선할 수 있는 구조로 만들어져 있다. 좀 더 직관적으로 이해하고 싶다면, generator를 지폐위조범, discriminator를 경찰이라 생각해 보면 된다.

> **지폐위조범(generator)은 위조 지폐를 최대한 진짜와 같이 만들어 경찰을 속이기 위해 노력하고, 경찰(discriminator)은 이렇게 위조된 지폐를 진짜와 구별(classify) 하려고 노력한다. 이런 과정을 반복하면서 두 그룹이 각각 서로를 속이고 구별하는 능력이 발전하게 된다. 궁극적으로는 지폐위조범의 솜씨가 매우 좋아져서 경찰이 더 이상 진짜 지폐와 위조 지폐를 구별할 수 없을 정도(구별할 확률 p=0.5)가 된다는 것.**

앞선 예시를 수학적 용어를 섞어 표현하면 다음과 같다. Generative 모델 G는 우리가 갖고 있는 실제 이미지 x의 분포(data distribution)를 알아내려고 노력한다. 만약 G가 정확히 데이터 분포를 모사할 수 있다면, 이 분포로부터 뽑은 (혹은 생성한) 샘플 이미지는 진짜 이미지와 전혀 구별할 수 없다.. 즉, G는 $z \sim p_z$일 때 가짜 이미지 샘플G(z)의 분포 p_g를 정의하는 모델로 생각될 수 있다. 여기서 z는 G라는 샘플링 모델에 들어갈 입력(input) 값이다. 보통 z의 분포 p_z는 가우시안(Gaussian) 분포를 사용하는데 이 부분은 차차 설명하겠다.

한편, discriminator모델 D는 현재 자기가 보고 있는 샘플이 진짜 이미지 x인지 혹은 G로부터 만들어진 가짜 이미지 G(z)인지 구별하여 샘플이 진짜일 확률을 계산한다. 앞서 예를 든 것처럼 $p_g = p_{data}$가 된다면 각각의 분포로부터 뽑힌 샘플만을 가지고 어느 쪽에서 왔는지 구별할 방법이 없기 때문에 D가 할 수 있는 최선은 '동전 던지기'이다. 즉, 임의의 샘플 x에 대해서는 [수식 1]처럼 나타낼 수 있다.

[수식 1]

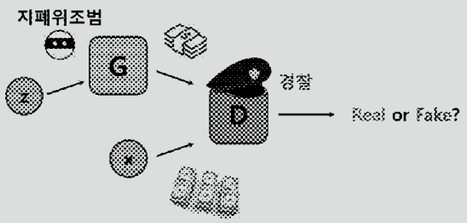
$$D(x) = \frac{1}{2}$$

최소최대의 문제 (minimax problem)

우리의 목적은 generator는 점점 더 실제 이미지와 닮은 샘플을 생성하게 하고 discriminator는 샘플을 점점 더 잘 구별하는 모델을 만드는 것이다. 따라서 [그림 3]을 보면 알 수 있듯이, D의 입장에서는 data로부터 뽑은 샘플 x는 D(x) = 1이 되고, G에 임의의 input z 넣고 만들어진 샘플에 대해서는 D(G(z)) = 0이 되도록

노력한다. 즉, D는 실수할 확률을 낮추기(minimize) 위해 노력하고 반대로 G는 D가 실수할 확률을 높이기(maximize) 위해 노력하는데, 따라서 둘을 같이 놓고 보면 "minimax two-player game or minimax problem"이라 할 수 있다.

[그림 3] 지폐위조범 VS 경찰



이를 수식으로 정리하면 우리가 하고자 하는 것은 다음과 같은 가치 함수(value function) V(G, D)에 대한 최소최대 문제(minimax problem)를 푸는 것과 같아진다.

[수식 2]

$$min_G max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1-D(G(z))]$$

무엇이든지 간에 이런 수식이 있으면, 극단적인 예시를 넣어 이해하는 것이 빠르다. 먼저 G 입장에서 가장 이상적인 상황을 생각해 보자. G가 진짜 이미지와 완벽히 닮은 샘플을 만들고, G(z)가 만들어 낸 이미지가 진짜일 확률이 1이라고 D가 잘못된 결론을 내린다면, D(G(z))=1이므로 두 번째 항의 값이 $-\infty$가 된다. 이 때 G의 입장에서 V가 "최솟값"이라는 것은 자명하다.

반면에 D가 진짜 이미지와 가짜 이미지를 완벽하게 잘 구별하는 경우, 현재 보고 있는 샘플 x가 실제로 data distribution으로부터 온 녀석일 때 D(x) =1, 샘플이 G(z)가 만들어 낸 녀석이라면 D(G(z))= 0이 되므로 위 수식 첫 번째 항과 두 번째 항이 모두 0으로 사라지죠. 이 때 D의 입장에서 V의 "최댓값"을 얻을 수 있다는 것 역시 자명하다.

이제 드디어 우리가 풀 문제를 수식으로 명확히 정의하였다. 그런데 보시다시피 이 문제는 변수 G와 D, 두 개가 서로 엮여 있다. 이런 관계 덕분에 서로에게 피드백을 줘서 성능이 각각 좋아지기도 하지만, 한편으로 한쪽 모델에 대해 문제를 풀면 다른 한쪽은 손해를 보기 때문에 둘 모두를 만족시키는 평형점을 찾기가 쉽지 않다.

따라서 실제로 문제를 풀기 위해서는 한쪽을 상수로 고정하고 다른 변수에 대해 문제를 푸는 방식의 접근을 할 수밖에 없다. 예를 들자면, 먼저 현재 생성 모델을 G'으로 고정하고, D와 관련해서는 다음 문제를 먼저 풀어서 D'을 구한다. [수식 3] 이렇게 얻은 D'를 넣고 G에 대해서는 다음 문제를 번갈아 푸는 전략을 쉽게 떠올려 볼 수 있다. [수식 4]

[수식 3]

$$\max_D V(D, G')$$

[수식 4]

$$\min_G V(D', G)$$

이론적 근거(theoretical results)

이제 방법도 알았고 문제를 열심히 잘 풀기만 하면 될 것 같지만, 그러기 전에 아직 해결해야 할 것들이 몇 가지 남아 있다. 먼저 우리가 정의한 이 문제가 실제로 정답이 있는지(existence), 만약 해가 존재한다면 유일한지(uniqueness) 확인할 필요가 있다. 그리고 제안한 방법이 실제로 원하는 해를 찾을 수 있는지(convergence) 확인해야 한다. 애초에 답이 없는 문제를 풀거나, 답이 있더라도 여러 가지이거나, 풀 방법을 제안했는데 그 방법이 해로 수렴한다는 보장이 없으면 여러 모로 곤란해진다.

따라서 이제부터는 이 부분들을 하나씩 체크해 보겠다. 이 장에서 중요한 내용은 크게 두 가지로 나눌 수 있다.

1) 최소최대 문제(minimax problem)가 $p_g = p_{data}$에서 전역해 (global optimum)로 유일해(unique solution)를 갖는다는 것과
2) 알고리듬이 global optimum로 수렴한다는 것이다.

일단, 첫 번째 주제인 global optimality에 관한 얘기를 하기 위해서는 먼저 가져가야 할 도구가 하나 있다. 임의의 G에 대하여, 최적의 discriminator D는 다음과 같다.

[수식 5]

$$D^*_G(x) = \frac{P_{data}(x)}{p_{data}(x) + p_a(x)}$$

위 수식을 증명하는 것은 크게 어렵지 않다. 먼저 V(G, D)를 다음과 같이 풀어 쓸 수 있다.

[수식 6]

$$\begin{aligned} V(G,D) &= E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1-D(G(z))) \\ &= \int_x P_{data}(x)\log(D(x))\,dx + \int_z P_z(z)\log\left(1-D(G(z))\right)dz \\ &= \int_x P_{data}(x)\log(D(x)) + p_g(x)\log(1-D(x))\,dx \end{aligned}$$

마지막 식의 형태를 참고하면, D*을 구하는 문제는 임의의 [수식 7]에 대하여.

[수식 7]

$$(a, b) \in \mathbb{R}^2 \setminus \{0, 0\}$$

[수식 8] 함수에서 최댓값을 구하는 문제로 단순화할 수 있다.

[수식 8]

$$y \to a log(y) + b log(1-y)$$

위 함수는 [수식 9]에서 최댓값을 갖기 때문에 이것으로 증명이 완료된다.

[수식 9]

$$y^* = \frac{a}{a+b}$$

결국, 위 결과를 바탕으로 최소최대의 문제(minimax problem)을 다시 써 보면 이제 변수가 G 하나인 C(G)라는 문제로 나타낼 수 있다.

[수식 10]

$$\begin{aligned} C(G) &= max_D V(G,D) \\ &= E_{x\sim p_{data}}[\log D^*_G(x)] + E_{x\sim p_z}[\log(1-D^*_G(G(z))] \\ &= E_{x\sim p_{data}}[\log D^*_G(x)] + E_{x\sim p_g}[\log(1-D^*_G(x)] \\ &= E_{x\sim p_{data}}\left[\log\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\right] + E_{x\sim p_g}\left[\log\frac{p_g(x)}{p_{data}(x)+p_g(x)}\right] \end{aligned}$$

전역적 최적해(global optimum) 증명

이제 이 도구를 사용해서 주요 정리(main theorem)을 증명해 보고자 한다. C(G)의 global minimum은 오직 $p_g = p_{data}$으로 유일하게 존재하며, 이때 C(G)의 값은 -log(4)이다.

증명은 양방향으로 진행된다. 먼저 가장 이상적으로 $p_g = p_{data}$ 일 때 C(G)가 어떤 값을 갖는지 확인하려 한다. 앞서 구한 [수식 5]에 $p_g=p_{data}$를 입력하면 [수식 11]이 나오고

[수식 11]

$$D^*_G(x) = \frac{1}{2}$$

이를 [수식 2]에 대입하면 다음과 같이 나타낼 수 있다.

[수식 12]

$$C(G) = \mathbb{E}_{x\sim p_{data}}[-log(2)] + \mathbb{E}_{x\sim p_g}[-log(2)] = -log(4).$$

한편, C(G) 수식을 바꿔 표현해서 -log(4)가C(G)가 가질 수 있는 유일한 최적값임을 알기 위해서는 다음과 같이 C(G)를 표현하는 것이 가장 중요하다.

[수식 13]

$$\begin{aligned} C(G) &= C(G) + \log(4) - \log(4) \\ &= -\log(4) + KL(p_{data} \,\|\, \tfrac{P_{data}+p_g}{2}) + KL(p_g \,\|\, \tfrac{P_{data}+p_g}{2}) \\ &= -\log(4) + 2\cdot JSD(P_{data} \,\|\, p_g) \end{aligned}$$

첫 번째 등호에서 두 번째 등호로 넘어가기 위해서는 쿨백-라이블러 발산(Kullback-Leible divergence)에 대한 정의를 알아야 한다. 쿨백-라이블러 발산은 P라는 확률 분포와 Q라는 확률 분포가 있을 때 두 분포가 얼마나 다른지를 측정하는 값이다.

[수식 14]

$$D_{KL}(P\|Q) = \sum_i P(i) log \frac{P(i)}{Q(i)}.$$

P와 Q의 분포가 일치하면 log 안의 값이 1이 되어 divergence가 0이 되는 것을 알 수 있다. 따라서 첫 번째 수식의 C(G)+log(4)를 풀어서, [수식 15]와 같이 나타낼 수 있다.

[수식 15]

$$E_{x\sim p_{data}}\left[\log\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\right] + E_{z\sim p_x(z)}\left[\log\frac{p_g(x)}{p_{data}(x)+p_g(x)}\right] + \log(2) + \log(2)$$

이와 같이 나타낸 후 각 기댓값(expectation) 안으로 log(2)를 넣어 주면 두 번째 등호의 수식 형태로 나오게 된다. 두 번째에서 세 번째 등호도 JSD(Jensen-Shannon divergence)의 정의를 알면 자연스럽게 따라온다.

[수식 16]

$$JSD(P \,\|\, Q) = \tfrac{1}{2} D_{KL}(P \,\|\, M) + \tfrac{1}{2} D_{KL}(Q \,\|\, M).$$

[수식 17]

$$M = \frac{1}{2}(P+Q)$$

여기서 [수식 17]이므로 그대로 원 수식에 비교 대조해 보면, 쉽게 유도할 수 있다. 결국 JSD는 분포 간의 차이를 나타내는 값으로 항상 양수이며, 비교하는 두 분포가 일치할 때만 값이 0이기 때문에 C*=-log(4)가 C(G)의 global minimum이며 그 유일한 해가 $p_g=p_{data}$임을 알 수 있다.

앞서 정의한 minimax problem을 잘 풀기만 하면(즉, 전역적 최적해를 찾으면), generator가 만드는 확률분포(probability distribution) p_g가 data distribution p_{data}와 정확히 일치하도록 할 수 있다. generator가 뽑은 샘플을 discriminator가 실제와 구별할 수 없게 된다는 것이다.

수렴성(convergence) 증명

남은 것은 앞서 제시한 것과 같이 G와 D에 대해 번갈아 가며 문제를 풀었을 때, $p_g = p_{data}$를 얻을 수 있는지 확인하는 것이다. 여기서 증명을 용이하게 하기 위해 몇 가지 조건에 대한 상정(想定)이 필요하다. 먼저 G와 D라는 모델이 각각 우리가 원하는 데이터의 확률 분포를 표현하거나 구별할 수 있는 모델을 학습할 수 있을 만큼 용량(capacity)이 충분하고, discriminator D가 주어진 G에 대해 최적값인 D*으로 수렴한다고 가정합니다.

위 가정이 충족되었을 때, [수식 18]에 대하여 G의 최적값을 구하면 p_g가 p_{data}로 수렴한다.

[수식 18]

$$\mathbb{E}_{x \sim p_{data}}[log D^*_G(x)] + \mathbb{E}_{x \sim p_g}[log(1 - D^*_G(x))]$$

먼저 D가 고정일 때, V(G,D)=U(p_g,D)라 표현하려 한다. 증명의 방향은 먼저 U(p_g, D)가 p_g에 대해서 볼록(convex) 함수임을 확인하는 것이다. U(p_g, D)의 유일한 해가 $p_g = p_{data}$라는 것은 앞선 정리에서 확인하였기 때문에 이로써 증명이 끝나게 된다. 정의에 의해 [수식 19]가 도출된다.

[수식 19]

$$U(p_g, D) = \int_x p_{data}(x)\, log\big(D(x)\big) + p_g(x)\, log\big(1 - D(x)\big)\, dx$$

p_g에 대해 미분을 하면 log(1-D(x))만 남게 되고 이는 p_g에 대해서 상수이기 때문에 U(p_g, D)가 선형 함수임을 알 수 있다. 그러므로 [수식 20]이 p_g에 대해 볼록 함수*로 유일한 전역해(global optimum)를 갖는다.

[수식 20]

$$\sup_D U(p_g, D)$$

따라서 p_g에 대한 적은 수의 gradient decent 계산만으로도*1 [수식 21]이 도출될 수 있다.

[수식 21]

$$p_g \rightarrow p_{data}$$

앞서 제안한 전략이 전역적 최적해로 해가 수렴한다는 것까지도 증명했다. 단! 지금까지 전개한 논리들에는 몇 가지 꼭 알고 넘어가야 할 내용들이 있다.

첫째, 지금까지 논리를 전개하면서 G와 D를 만들 모델에 대해 어떤 제약을 두지 않았다. 즉, 여기서 G와 D는 무한대의 표현 용량을 가진(임의의 어떤 함수라도 표현할 수 있는) 모델로서 앞서 진행한 증명들은 모두 확률 밀도 함수(probability density function) 공간에서 전개된 것이다. 그러나 실제로 모델을 만들 때 위와 같은 non-parametric 모델을 만드는 것은 현실적으로 어려움이 많다.

그런데 뉴럴 네트워크은 충분한 용량만 주어진다면 보편 근사기(universal function approximator)로서 역할을 잘해 주며, 미분이 가능한 함수들로 표현되므로 역전파(backpropagation)라는 매우 직관적이고 효율적인 학습이 가능하다. 또한 뉴럴 네트워크은 확률분포 (probability distribution)를 학습하기 위해 마르코프 사슬 몬테 카를로(Markov Chain Monte Carlo, MCMC)와 같은 다소 소모적인 방법을 사용하지 않아도 되기 때문에 상대적으로 빠르다. 이러한 여러 장점들 때문에 실용적인 측면에서 GAN에서는 뉴럴 네트워크을 활용한다.

각각의 G와 D를 단순한 뉴럴 네트워크인 다중층 인식망(multilayer perceptron, MLP)으로 모델을 만들면, G는 입력값(input) z에 대해 출력값(output)으로 이미지 샘플을 뽑아 주는 G(z; θ_g)로 표현할 수 있다. 한편, D 역시 D(x; θ_d)로 나타내며 output으로 샘플 x가 p_{data}로부터 뽑혔을 확률값을 내보낸다. 여기서 θ_g와 θ_d는 각 MLP의 매개변수(parameter)이다.

그러나 이렇게 함수 공간을 일반적인 공간에서 뉴럴 네트워크이 표현할 수 있는 공간으로 바꾸게 되면, GAN이 표현할 수 있는 p_g가 G(z;θ_g) 함수로 나타낼 수 있는 종류로 제한이 된다. 뿐만 아니라 더 이상 p_g에 대해 직접 최적화 문제를 푸는 것이 아닌 θ_g에 대해 최적화 문제를 푸는 것으로 모든 구조가 바뀌기 때문에 앞서 증명들에서 사용한 가정들이 모두 깨지게 된다. 즉, MLP를 모델로 사용하면 G가 모수 공간(parameter space)에서 다중임계점(multiple critical point)을 가지기 때문에 완전히 증명한 바와 합치하지는 않는다. 다만 실제로 학습을 해 보면 성능이 잘 나오는 것으로 미루어 봤을 때, MLP가 이론적 보장이 좀 부족할지라도 실용적으로 쓰기에는 합리적인 모델이라고 할 수 있다.

둘째, 수렴성 증명에 들어간 가정들은 '생각보다' 매우 강력한 제약이다. 실제로는 계산량이나 학습 시간에 한계가 있기 때문에 D*를 얻기 위해 D가 수렴할 때까지 학습시키는 것은 현실적으로 어렵다. 게다가 모델의 용량이 p_{data}를 학습하기에 충분한지에 대한 문제 역시 다루기 어렵다. 최대한 이론적 상황과 비슷하게 맞춰 주기 위하여 D에 대한 학습을 G에 비해 보다 많이 수행해 주는 등 기술적인 노하우를 사용하기는 하지만 근본적인 해결책이라 할 수는 없다. 따라서 어떤 면에서는 유명무실한 증명이라 할 수도 있다.

셋째, 심지어 구현 시 사용하는 가치 함수(value function)의 형태가 이론과 다르다. 실제로는 $\min_G$ log(1-D(G(z))) 대신 $\max_G$ log(D(G(z)))로 모델을 바꾸어 학습시킨다. 학습 초기에는 G가 매우 이상한 이미지를 생성하기 때문에 D가 너무도 쉽게 이를 진짜와 구별하고 log(1-D(G(z)))의 기울기가 아주 작아서 학습이 엄청 느리다. 문제를 G = $\max_G$ log(D(G(z)))로 바꾸면, 위와 같은 문제가 생기지 않기 때문에 쉬운 해결 방법이다. 하지만 문제를 다른 형태로 바꿨기 때문에 이 역시도 아쉬운 점이라 할 수 있다.

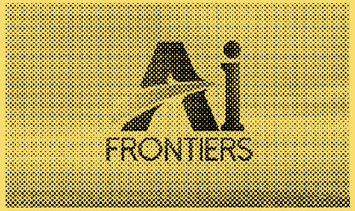
이로써 GAN에 대해 원 논문에서 나온 이론적 증명까지 모두 살펴보았습니다. 다음 호에서는 이런 배경지식을 바탕으로 기존 생성 모델과 GAN의 차이점에 대해 살펴보고 GAN의 특징과 장단점에 대해 본격적으로 소개하겠다.

*1 참고 | P_g에 대해 볼록함수이기 때문에 이 함수의 최소값을 구하는 것을 gradient descent로 iterative하게 찾으면 적은수의 업데이트 만에 최소값 P_g를 찾게 된다.

국내·외 AI 컨퍼런스 소개

12월에 열리는 NIPS로 전 세계 AI 연구자들의 관심이 쏠려 있지만, NIPS 외에도 여러 곳에서 11월과 12월에 학회가 열립니다. 국내외에서 열리는 AI 학회를 몇 곳을 정리, 공유합니다. 해외 학회에 직접 참여가 어려우신 분들은 해당 행사의 소개 자료를 담고 있는 홈페이지에라도 들어가서 최신 동향을 확인해 보시는 건 어떨까요?

AI FRONTIERS CONFERENCE



'AI Frontiers Conference'는 인공지능과 빅데이터 전문가들을 대상으로 진행되는 컨퍼런스입니다. 자연어처리/ 자율주행 자동차 / 챗봇 / 로봇 / 딥러닝 / 영상 분석 / 게임 등의 주제가 이번 학회에서 논의될 예정입니다.

기간 | 11월 3일 ~ 11월 5일
장소 | 미국 산호세
링크 | http://aifrontiers.com

ODSC West 2017

'ODSC West 2017'은 데이터 과학 컨퍼런스입니다. 오픈 소스 툴, 라이브러리, 언어 연구의 전문가들이 발표자로 참여합니다. 이번 학회의 핵심 주제는 딥러닝 / 예측 분석 / 자연어 처리 / 데이터 시각화 / 인지 컴퓨팅 / 데이터 랭글링 등을 다룹니다.

기간 | 11월 2일 ~ 11월 4일
장소 | 미국 샌프란시스코
링크 | https://odsc.com/california

AI World

'AI World'는 기업차원에서 인공지능과 기계학습을 다루는 컨퍼런스입니다. 인공지능과 기계학습을 이용해 새로운 사업 기회를 발굴하고 비용 감소에 대해서 논의하는 자리가 될 것입니다. AI 헬스케어/ 가상비서/ 기계학습과 딥러닝/ 인지컴퓨팅 등 다양한 주제를 다룰 예정입니다

기간 | 12월 11일 ~ 12월 13일
장소 | 미국 보스톤
링크 | https://aiworld.com

DataSciCon.Tech

'DataSciCon.Tech'는 데이터 과학의 전 세계 전문가들이 모여 최신 연구를 논의하는 장입니다. 이번 학회의 주제는 크게 4가지입니다. 주제는 발견과 혁신 그리고 가치 창출을 위한 데이터 분석 / R을 이용한 데이터 과학 / 파이썬(python)과 텐서플로우(tensorflow)를 이용해 배우는 기계 학습 입문 과정 / 태블로(tableau)를 이용한 데이터 분석입니다.

기간 | 11월 29일 ~ 12월 22일
장소 | 미국 아틀랜타
링크 | http://www.datascicon.tech

AI Europe 2017

'AI 유럽 2017'은 2016년 유럽에서 처음으로 열렸습니다. 첫 학회의 주제는 '경영에 적용하는 AI'였습니다. 이번에 두 번째로 열리는 'AI 유럽 2017'의 주제는 다음 세 가지입니다. 자연어처리 시스템을 이용해 고객과 파트너의 경험을 강화시키는 방법 / AI와 인지 솔루션 그리고 로보틱스 프로세스의 자동화 / 빅데이터와 딥러닝을 결합해더 나은 의사결정을 내리는 방법

기간 | 11월 20일 ~ 11월 21일
장소 | 영국 런던
링크 | https://ai-europe.com

MLconf 2017

'MLconf(Machine Learning Conference)'는 2012년 시작된 학회입니다. 이 학회는 그래프 데이터베이스(Graph Database)에 집중합니다. 규모가 크고, 노이즈(noise)가 많은 데이터 셋(data set)이 결합된 난해한 문제들을 해결하기 위해 필요한 플랫폼과 툴, 알고리듬 그리고 최근의 연구들이 이번 학회에서 논의될 예정입니다.

기간 | 11월 10일
장소 | 미국 샌프란시스코
링크 | https://goo.gl/p9fFQd

The AI Summit New York

'The AI Summit New York'은 AI가 기업 조직에 미치는 실질적 영향 및 비즈니스 생산성에 일으키는 변화 등 실용적 솔루션에 초점을 맞춘 컨퍼런스입니다. 아마존 알렉사(Amazon Alexa), IBM 왓슨(Watson), 마이크로소프트 등이 파트너로 참여하는 이번 컨퍼런스에서는 AI 생산성 / 머신러닝 / AI 규제 등 다양한 주제에 관해 약 50개 이상의 세션이 진행될 예정입니다.

기간 | 12월 5일 ~ 12월 6일
장소 | 미국 뉴욕
링크 | https://theaisummit.com/newyork/

NVIDIA 딥러닝 데이 2017

GPU 개발사 엔비디아(NVIDIA)에서 딥러닝 기술 트렌드와 딥러닝 적용 사례를 소개하는 컨퍼런스를 개최합니다. 구체적으로는 딥러닝 / 헬스케어 / 인공지능 / 자율주행 / 인공지능 스타트업의 소주제 트랙이 구성되어 있습니다. 또한 GPU 기반의 딥러닝 프레임워크를 활용해 실습을 해보는 세션도 함께 제공합니다.

기간 | 10월 31일 ~ 11월 2일
장소 | 서울 코엑스
링크 | https://www.nvidia.com/ko-kr/deep-learning-day/agenda/

2017년 인공지능 컨퍼런스

'한국인공지능학회'에서 2017년 인공지능 컨퍼런스를 개최합니다. 한국인공지능학회는 인공지능과 관련된 연구, 산학연 기술 협력을 목적으로 2016년 말 설립되었습니다. 컨퍼런스에는 의료 / 국방 / 영상 인식 / 로보 어드바이저 등 다양한 분야의 전문가들이 참여할 예정입니다.

기간 | 11월 3일
장소 | SKT 타워 4층 SUPEX Hall
링크 | https://goo.gl/pCAix9

마치며

어느 새 부터인가 인공지능(artificial intelligence, AI)을 빼고는 미래를 논하기 어려운 시대가 된 듯 합니다. 현실을 진단하고 미래를 내다보려는 목적 모임에서는 규모의 크기와 상관없이, 인공지능을 논하곤 합니다. 최근 국내에서 열린 한 포럼에서 AI 전문가들의 고견을 들을 기회가 있었습니다. 국내외 전문가들의 이야기를 전달드리면서, 이번 호를 맺음하고자 합니다(아래 저명 인사들의 말을 명확하게 하기 위한 편집 작업이 일부 진행됐음을 알려 드립니다. 편집 과정에서 본 의도의 훼손 및 변경이 없도록 했습니다.)

"AI는 우리 사회의 본질을 바꿀 변화를 이끌 수 있는 가장 강력한 요소이다. AI 기반으로 전혀 다른 산업이 출현할 것이다. 그러나 자의식을 가진 AI의 출현은 일어나지 않을 것이다. 현재 AI의 주류 연구는 사람을 돕는 기술에 초점을 두고 있으며, AI 자의식 개발과는 거리가 멀다. 개인적으로 AI보다는 지능형 대리인이라는 뜻의 IA(intelligent agent)라는 표현이 적확하다고 생각한다." (장 야친 바이두 총재)

"소셜네트워크나 미디어는 인공지능의 위협에 대해 크게 과장한다. 인공지능의 위협 중 하나는 혜택이 소수의 사람에게만 집중되어 불평등을 야기할 수 있다는 것이다. 그렇지만 민주주의 사회가 지속된다는 가정하에서 인공지능이 가져올 미래는 오히려 낙관적이라고 생각한다. 또한 1,000억개가 넘는 뉴런을 가진 인간의 뇌를 인공지능이 따라잡는 건 먼 미래의 일이라고 생각한다." (이대열 예일대 의학대학원 신경과학과 교수)

"장기적 관점에서 '초지능(Super Intelligence)'사회의 도래는 인간의 존재 및 실존주의를 위협할 것이다. 인공지능이 인간 뇌를 모방하는 방식을 따르지 않더라도 기계적 알고리듬을 통해 초지능을 가질 수 있다. 뇌 속에 1,000억개의 뉴런이 있지만 인간의 신경속도는 기계의 정보 전달 속도를 따라잡을 수 없다. 인공지능에는 어떤 공간적 제한도 없다는 것을 유념해야 한다. 또한 소수의 자본가 독재 세력이 인공지능을 독점하여 세계를 지배하는 것을 경계해야 한다." (닉 보스트롬 옥스퍼드대학교 인류미래연구소 소장)

"인공지능을 이용해 여론을 완전히 이해하는 건 굉장히 어려운 일이다. 인공지능이 특정 데이터와 알고리듬만을 이용하여 인간을 이해하려 한다면 더더욱 인간을 이해할 수 없을 것이다. 데이터보다 인간의 의도를 우선시 해야 이러한 문제를 해결할 수 있다." (제이피 클로퍼스 브랜드아이 CEO)