

KAKAO

Vol.06

AI

2017.08

AI CODE

```
__global__ void compute_sentence_pzx (
    float *pzx, float *mat_alpha, float *mat_beta,
    int *frame_num , int max_frame_len ,
    int *label_len, int max_label_len, int num_sentence,
    int nTotalProcess
)
{
    int sent_idx = threadIdx.x + ( blockIdx.x * blockDim.x );
    int s, t ;

    if( sent_idx < nTotalProcess ){
        sent_idx = sent_idx % num_sentence ;
        t = frame_num[ sent_idx ] - 1;
        s = [ num_sentence * t + sent_idx ] * max_label_len + label_len[ sent_idx ];
        pzx[ sent_idx ] = LogAPlusB( mat_alpha[ s - 1] , mat_alpha[ s - 2 ] );
    }
}

__global__ void add_source_err ( float *cell_err_tmp, float *cell_err, int Nc, int Ni, int N )
{
    int idx = threadIdx.x + ( blockIdx.x * blockDim.x );
    int idx2, r1, d1, Np;

    if( idx < N ){
        Np = Ni + Nc;  d1 = idx % (Np+1) ;
        if( d1 > Ni && d1 < Np ){
            r1 = idx / (Np+1) ;  idx2 = r1 * Nc + (d1 - Ni);
            cell_err[ idx2 ] += cell_err_tmp[ idx ];
        }
    }
}
}
```



KAKAO AI REPORT

Vo1.06

발행일 | 2017년 8월 31일

발행처 | (주)카카오

발행인 | 카카오 정책지원파트

편집 | 김대원, 양현서, 정수현

디자인 | 허진아

메일 | kakaoireport@kakaocorp.com

COVER

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

Vol.06 코드 | 김훈 ekdon.kim@kakaocorp.com

음성인식 엔진의 음향모델을 학습하기 위한

알고리즘 일부를 발췌한 것입니다. 음향모델은

음성신호의 특징을 추출하여 모델링하는 것으로

딥러닝 기술을 이용하여 학습합니다. 음성인식은

다음앱, 카카오맵, 카카오내비 의 음성검색에

적용되어 서비스 중입니다.

contents

preface 02

industry

데이터와 음성인식

하용호 | 머신러닝 적용의 실제: 논문이 가르쳐 주지 않는 것들 06

김훈 | 음성인식 방법과 카카오의 음성형엔진 **AI in Kakao** 12

trends

생생한 AI 현장의 이야기

이상용 | 국제인공지능법학회 참관기 20

하영식 | AI 접목된 의료영상의 주요 플레이어들 26

learning

AI 연구 동향과 강화학습 개념

정수현 | AI 3대 학회의 발표 논문 경향 32

엄태웅 | 딥러닝 연구의 현재와 미래 part 2 36

최성준, 이경재 | 알파고를 탄생시킨 강화학습의 비밀 part 2 42

information

앤드류 응의 코세라 딥러닝 전문가 과정 소개 50

closing 52

카카오 AI 리포트 6호를 내며

6번째 카카오 AI 리포트가 발행됐습니다. 이번 호에서는 피상적으로 알고 있는 소재에 대한 접근을 우선적으로 해봤습니다. AI 연구와 산업에서 “data is king”이란 말이 통용될 정도로, 데이터의 값어치는 높게 평가됩니다. 허나, “데이터는 다다익선(多多益善)이란 명제가 항상 진리일까요?” 이 질문을 카카오 Machine Intelligence 파트의 하용호 파트장에게 해봤습니다. 하 파트장이 현장에서 AI를 적용하면서 데이터와 관련해 겪은 실제적 이야기를 정리해 주셨습니다.

일반인이 접하게 된 AI가 접목된 대표적 서비스 중 하나가 음성인식입니다.

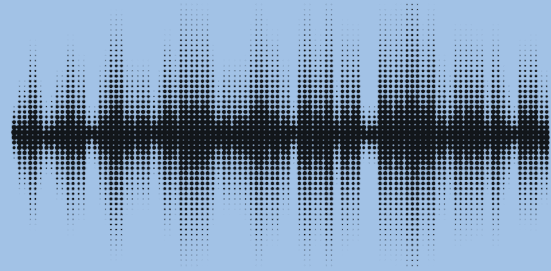
음성인식이 어떻게 이뤄지는지에 대한 쉬운 설명이 담긴 글은 많지 않습니다. 그래서, 카카오 AI기술팀에게 글을 청탁했습니다. 현업의 일만으로도 눈코 뜰 새 없는 분들이기에, 부탁 그 자체에 의미를 두고 글을 요청드렸습니다. 김훈 음성처리파트장이 소상한 설명이 담긴 글을 주셨습니다. 원고에는 다음 달 15일 출시되는 신차인 제네시스 G70에 탑재되어 화제가 된 카카오의 음성형엔진에 대한 설명도 담겨 있습니다.

이상용 충남대 로스쿨 교수님의 런던 국제인공지능법학회 참관기, 그리고 하영식 바임컨설팅 파트너님이 써주신 AI가 접목된 의료영상 업계 이야기에는 현장감이 가득합니다. 엄태웅 님의 딥러닝 연구 논문 소개와 저희 편집진의 정수현 님이 정리해 주신 올해 열린 3대 AI 학회 연구 논문 분석 결과는 AI연구의 어제와 오늘을 간명하게 훑고자 하는 분들에게 도움이 되실 겁니다. 지난 6월 발행된 카카오 AI 리포트 4호에 발행되어 큰 관심을 모았던 최성준 님과 이경재 님의 ‘알파고를 탄생시킨 강화학습의 비밀’의 두 번째 글이 이번 호에 게재됩니다.

바이두를 떠난 뒤에 작은 행보도 주목받고 있는 앤드류 응 교수의 코세라 딥러닝 전문가 과정의 주요 내용을 끝으로 이번 호를 마무리했습니다. 이번 호도 귀한 시간을 내주신 독자 분들에게 가치있는 콘텐츠로써 역할을 하길 고대합니다.

2017년 8월 31일
카카오 정책지원파트 드림

데이터와 음성인식



데이터와 음성인식은 AI가 화두가 된 이후에 키워드로 손꼽히는 단어입니다. 데이터는 AI 모델의 경쟁을 좌우하는 요소로써, 음성인식은 AI가 발현된 대표적인 결과물로써 주목을 받고 있습니다. 그러나, 데이터가 현장에 적용되는 과정, 그리고 음성인식 방법에 대한 구체적인 설명이 소상히 담겨져 있는 콘텐츠는 드뭅니다. 데이터를 중심으로 서비스에 AI를 적용했을 때 겪을 수 있는 문제, 그리고 그것을 해결하기 위한 방법에 대한 설명이 첫 번째 글에 담겨 있습니다. 두 번째는 음성인식 방법에 대한 상세한 소개와 현재 카카오 미니(Kakao mini) 그리고 현대자동차의 제네시스 G70에 탑재되며 세간의 주목을 받고 있는 카카오의 음성형엔진 일부에 대한 설명이 포함되어 있습니다.

머신러닝 적용의 실제: 논문이 가르쳐 주지 않는 것들

누가 뭐라고 해도 역시 지금의 슈퍼스타는 머신러닝(machine learning)이다. 마법같은 결과를 담은 논문이 나날이 쏟아져 나오고, 또 며칠만 있으면 코드로 구현되어 깃허브(github)를 통해 모두에게 공유된다. 거기다 세상에는 친절할 사람들도 어찌나 많은지, 쉽게 쓰여진 좋은 튜토리얼과 블로그 글들이 가득하다. 공부를 하기에 이보다 더 좋을 수 없다. 새로 산 GPU가 달린 머신 위에 ‘주피터 노트북(jupyter notebook)’¹⁾ 을 띄워놓고, 정확도를 1%씩 올리는 과정이 ‘꿀잼’이다. 그런데 이런 즐거움이 현실을 만나서 조금 고생을 해야할 때가 있다. 공부로 끝나는게 아니라 실제 서비스에 머신러닝을 적용해야 할 때가 바로 그 순간이다. 오늘은 논문들이 가르쳐주지 않는 현실의 고민들을 함께 알아보자.

애초에 고민할 일이 왜 일어나는가?

높은 정확도를 가진 모델을 잘 만드는 것으로 부족한가? 애초에 왜 고민을 해야 할까? 원인은 연구와 사업의 입장 차이 때문이다. 두 관점에서 보는 ‘잘하고 있다’의 기준이 다르다. 실제 서비스에 적용될 때는 연구자의 시선보다 사장님의 시선으로 고민을 해야한다. 사장님의 시선이란 뭘까? 정확도가 좋은 것도 좋지만, 결과적으로 효용을 발생시키고, 지속시키는 것을 중요하게 여긴다. 좀 더 쉬운 말로 풀자면 ‘사용자가 행복해 하는가?(사용자 만족)’, ‘그래서 돈이 되는가?(수익성)’, ‘계속 돈이 되는가?(지속성)’이다. 이렇게 바뀐 기준을 만족시키려다 보니, 노트북에서 퍼센트를 찍어낼 때와는 사뭇 다른 관점에서 고민해야 한다. 어떤 것들이 있고 왜 고민해야 하는지 탐색해보자.

모든 것은 변한다, 고로 계속 지켜봐야 한다.

머신러닝을 적용하는 과정을 상상해보면 다음과 같을 것 같다.

【 머신러닝을 현실에 적용하는 과정 】
1. 대용량의 데이터 스냅샷(snapshot) ²⁾ 을 받는다.
2. 열심히 모델을 만든다
3. 오~ 나름 맞추는 것 같다.
4. 적절히 API를 붙이고 서비스에 적용한다.
5. 성공! 그리고 다른 일을 하러간다.

하지만 이것으로 끝일까? 아니다. 세상 모든 것이 변하듯 서비스도 그렇다. 서비스가 변하면 모델링도 바뀌어야 한다. 이러한 변화는 서비스 내적인 요인과 외적인 요인으로 나눌 수 있다.

서비스 내적인 변화의 대표적인 예는 ‘사용자의 노후화’이다. 예를 들어서 오픈한지 갓 한달된 서비스를 상상해보자. 서비스 오픈 시에 자연 유입되는 사용자들은 일반과 대비해 얼리어답터(early adopter)³⁾ 성격이 강하여, 상대적으로 더 탐색적이고, 더 호응이 좋다. 이 때 모델을 만든다면, 모델에는 사용자의 적극적 성향이 크게 반영될 것이다. 그런데 만약 오픈한지 5년쯤 지난 서비스라면 어떨까? 신규 유입은 줄고, 대부분의 사용자는 잔존 사용자(revisiting user) 일 것이다. 이들은 서비스에 대해 이미 알만큼 알고, 조금은 지루해 한다. 이 때 학습시킨 모델은 초기 모델과는 분명 다른 특성을 보일 것이다. 이처럼 시간이 지나면 사용자 구성도, 서비스 사용 패턴도 변화하는데, 의외로 많은 모델이 한번 구워지고, 꽤 오래 방치된다. 정지된 모델은 낡아질 뿐이다.

서비스 외적인 변화는 주로 외부의 이벤트들로 부터 야기된다. 늘 일어나는 외부 이벤트로 대표적인 것이 주기성(seasonality)이다. 예를 들어 우리가 쇼핑몰을 운영한다 치면 여름에 학습시켰던 모델은 얇고 가벼운 옷을 상대적으로 더 추천할 것이다. 겨울에 학습시켰던

모델은 두꺼운 점퍼에 치우침(bias)을 가지고 있을 것이다. 여름에 만들었던 모델로 겨울까지 쓰고 있다 생각하면 아찔하다. 그 외에도 어떤 가수의 패션이 대유행을 한다던가 하는 다양한 외부 사건들이 사용자의 행동을 바꾼다. 이런 부분까지 반영되려면 모델은 계속 업데이트 되어야 한다.

다들 서비스에 적용할 때, 시스템 오류 모니터링까지는 잘 하곤 한다. 하지만 방치된 모델은 오류 만큼이나 해롭다. 머신러닝을 서비스화 시킬 때는 성능(performance) 모니터링과 그에 따른 재모델링 자동화(model rebuild automation)까지 연동시키는 것이 작업의 완성이다.

많은 데이터보다 신선한 데이터가 낫다.

우리의 데이터에 대한 사랑은 조금 맹목적이어서, 데이터가 많으면 많을수록 무조건 좋다고 믿는다. 대체로 맞지만, 때로 틀리다. 데이터에도 유효기간이 있기 때문이다. 일반적으로 모델의 학습 과정에서 전체 데이터를 트레이닝 셋(training set)과, 테스트 셋(test set)으로 분리한다. 트레이닝 셋 데이터를 통해 모델은 현실의 케이스를 학습 한 후, 테스트 셋 데이터를 사용하여 처음 보는 데이터도 잘 추론 하는지를 확인한다. 일반적으로 모델은 트레이닝 과정에서 과적합(overfitting)⁴⁾또는 과소적합(underfitting)⁵⁾하기 쉽기 때문에, 모델의 정확도는 반드시 테스트 셋(test set)를 통해서 평가되어야 한다. 여기까지는 모두가 잘 아는 사실이다.

그런데 이런 데이터 셋들을 만드는 전략이 문제이다. 보통의 경우는 모을 수 있는 과거의 데이터를 모두 다 끌어모아서 크게 만든 후, 이를 무작위(random)로 분리하여 트레이닝과 테스트를 만든다. 하지만 예를 들어 3년전의 데이터로 학습한 모델이, 지금 가입하는 사용자를 평가하기에 적절할까? 반대로 이번달 가입한 사용자로 학습된 모델이 3년전 데이터가 포함된 테스트 셋으로 평가되는게 합당할까?

시간에 불변한 서비스를 하는 기업이라면 상관없겠지만, 그런 회사는 별로 없다. 우리 모델이 실제 적용됐을 때, 테스트 셋만큼의 성능이 안나오는 것은 애초에 테스트 셋이 ‘지금의 사용자’를 제대로 반영하지 않은 데 따른 결과일 수 있다. 데이터도 시간이 지나면 상하기 때문에, 데이터셋 운영 전략이 함께하여야 좋은 성능을 유지할 수 있다. 먼저 무조건 큰 데이터보다는 데이터 양이 학습에 충분하다면 과거는 제한하여 신선한 데이터를 유지하는 것이 필요하다. 또 테스트셋은 상대적으로 최근 데이터에 가중치를 두고 뽑아 ‘지금 시기에 잘 작동(working)하는가’에 중점을 두도록 변경되어야 한다.

글 | 하용호 henry.ha@kakaocorp.com

데이터 사이언티스트로 일해왔습니다. 다사다난한 과정을 거쳐왔습니다. 현실과 기술, 그 두 개의 끈을 맞달아 매듭짓는 일을 해 나가고 싶습니다.

열심히 일한 탓에 데이터가 편향되어 버린다.

미국 드라마 '닥터 하우스'의 괴짜 의사 하우스는 환자 병의 원인을 알기 위해, 현재 받고 있는 모든 치료를 일부러 끊어버린다.

해열제도 진통제도 맞지 못하는 환자는 괴로워하지만, 오히려 이런 상황이 되고나서야 아픈 곳이 명확히 드러나고, 원인이 발견되어 치료가 가능해진다. 그동안 환자가 먹고 있던 약이 오히려 증상을 가리는 역할을 하고 있었던 것이다.

데이터에 기반한 액션을 오래 하다보면 비슷한 아이러니(irony)가 생겨난다. 실무에 적용하기 이전에는 사용자 집단 전체에 대해서 모든 사례의 데이터를 고르게 가지고 있지만, 실무에 적용하고 나서부터는 모델의 유도를 당한 사용자의 데이터만 가지게 된다. 즉 우리의 적극적 액션이 데이터의 편향(bias)을 만들어내고 편중(variance)을 만들어낸다. 다양성(diversity)이 훼손되는 것이다. 이러한 일반성 훼손은 변화를 눈치채지 못하게 한다.

예를 들어 '가나다' 라는 콘텐츠가 있었다고 해보자. 최적화 전에는 이 콘텐츠가 모두에게 노출이 되어 중립적 데이터를 만들어 내고 있었다. 그리고 이 데이터를 기반으로 모델이 A,B,C 사용자 집단 중에 A가 가장 반응이 잘한다는 사실을 찾아주었다. 그 이후 모델은 적극 사용되고 이 콘텐츠는 A 위주로만 노출된다. 문제는 시간이 지나보니 예전엔 아니었던 C가 더 반응을 잘하게 되었다던가, 우리가 몰랐던 좋은 D가 신규로 유입되었다던가 하는 일이 물밑에서 일어나도 우리가 눈치를 챌 수가 없다는 것이다. 동작하고 있는 모델 때문에 C나 D에게 콘텐츠가 노출되지 않기 때문이다. 이러한 일을 방지하기 위해, 우리는 모델이 적용되지 않아, 행동의 다양성을 보유한 무작위(random) 데이터를 일정량 이상 확보할 필요가 있다. 사실 이것은 데이터 기반 최적화의 전통적인 딜레마인 탐색과 획득(explore vs exploit) 트레이드 오프(trade off) 문제이다. 빼먹는데(exploit)에 집중하면, 변화를 위한 새로운 먹거리 발견(explore)이 미흡해진다.

물론 이 트레이드 오프 사이의 균형을 자동으로 찾아주는 멀티암드밴딧(multi armed bandit, MAB)⁶ 같은 알고리즘이 있기는 하다. 하지만 MAB의 목적은 '여러 모델간의 경쟁 우위'를 최적화 하기 위한 것이지, 무작위 데이터 확보를 위해서 쓰이는 것은 아니다. MAB 알고리즘에서 무작위 데이터를 얻기위해서 '아무것도 하지 않는 모델'을 추가시킬 경우, 이 모델은 성능 자체는 매우 낮은 값을 가질 것이며 MAB의 알고리즘 특성상 이런 모델은 아주 적은 노출 기회를 가지게 된다. 이 정도 적은 양으로는 위에서 기술한 변화를 감지하기 어렵다. 따라서 일시적으로 전체 트래픽을 처음부터 분리해 일정부분은 모델없이 노출시키고, 나머지는 MAB로 보내도록 하는 데이터 분리가 필요할 수도 있다. 물론 이 작업은 늘 이루어질 필요는 없고 분기나, 해마다 한번도 충분하다.

데이터는 자연적으로 중립성있게 생산되는 것이 아니라, 우리가 유도하는 방향으로 굴곡져 생성된다는 사실을 늘 기억하자. 의도적으로 얻어낸 다양성이 보존된 데이터는 우리가 놓친 기회는 없는 지를 확인하는 데 큰 도움이 된다.

비용에 따라 모델의 정확도보다 커버리지가 우선되기도 한다.

우리 제품을 구매할 것 같은 사용자를 찾는 타겟팅(targeting) 문제를 머신러닝으로 풀고 있다고 가정해보자. 모델을 고도화 시키고 정교화 시킬수록, 정확도(accuracy)는 올라가겠지만, 동시에 이 모델에 통과되기 위한 기준값(threshold)이 올라가면서 부작용으로 모델이 맞출 수 있는 대상의 수(coverage)가 작아질 수 있다. 이는 역으로 생각해보면, 너무 엄격한 기준을 적용한 나머지, 어쩌면 우리 서비스를 위해 지갑을 열었을지도 모르는 사용자를 놓치는 형상이 될 수 있다.

이럴 때 비즈니스 관점에서 보면, 정확도를 낮추더라도 더 많은 소비자가 포함될 수 있게 타겟의 범위를 넓히는 것이 합리적인 경우가 있다. 간단한 예를 들어 보자. 사용자의 구매 전환율은 10%이며 1만명의 메일 대상자를 선정해주는 모델A와 구매전환률이 5%이지만 10만명을 추출해주는 모델B가 있다고 해보자. 모델 A는 10% x 1만명으로 1000명의 구매를 만들겠지만, 모델 B는 5% x 10만명으로 5000명의 구매를 만들어낸다.

의료와 같이 잘못된 판단이 치명적인 분야라면 정확도를 기준으로 모델을 골라야 한다. 하지만 마케팅이나 푸시등 오판에 대한 비용이 적은 분야는 커버리지가 우선시 될 수 있다. 기술은 성능(performance)를 추구하지만, 비즈니스는 이익(profit)을 추구하기 때문이다.

서비스를 위해 정확도보다 속도를 더 우선하기도 한다.

모델 서비스 시에, 미리 다 계산해두는 배치(batch) 형식이 아니라 아니라, 사용자 요청 때마다 계산되는 온라인(online) 형태로 진행되어야 할 때가 있다. 사용자가 어떤 입력값을 넣을지 그 자유도가 너무 높아서 적절한 예상 답안을 미리 구워두기 힘든 경우가 그렇다.

배치는 대량의 데이터를 미리 계산해두고 서비스 특성에 맞는 적당한 자료 서버에서(hbase, redis, mysql등) 불러오기만 하면 되지만, 온라인은 훨씬 더 많은 고민이 필요하다. 그 고민 중 하나가 응답 시간(response time)이다. 사용자가 어떤 입력을 넣을지 예측이 어려워서 채택된 온라인이므로, 태생적으로 두껍고 긴 롱테일(long tail)을 가지게 된다. 두터운 롱테일은 이전의 계산 결과를 재활용하기가 어렵다는 뜻이므로 캐시(cache)도 소용없어서,

바로바로 계산해야 하는 양이 많다. 트레이닝(training)에 비하면 결과값 얻기(predict)는 몇천배 빠르겠지만, 복잡한 모델을 빌드 했을 경우에는 이 계산 속도도 문제가 된다.

문제가 되는 대표적인 예가 RNN계열(LSTM, GRU 등등)인데 애초에 재귀적(recursive)으로 계산되기 때문에 병렬화가 느린 RNN은 이미 구워진 모델이라 할지라도 건당 계산(predict)이 느리다. 머신러닝의 결과물이 최종결과가 아니라, 상위 큰 시스템의 구성요소로 쓰이는 경우가 대부분이기 때문에 서비스에 따라 20ms 이하의 아주 극단적으로 빠른 처리 속도를 요구받곤 한다. 느린 RNN계열은 이 요구 조건을 못 맞추는 일이 발생할 수 있다. 이럴 때는 정확도가 조금 낮더라도 같은 일을 하는 CNN 구조가 있다면 그 편이 더 좋다. 병렬화가 쉬운 구조적 특성 때문에 CNN계열은 RNN계열보다 빠르다. 정확도는 1% 낮지만 0.1초 더 빨리 서빙해서 사용자에게 빠릿빠릿한 감각을 맛보여 준다면, 사용자에게 주는 총체적 효용은 후자가 더 클 수 있다. '번역'과 같이 사용자가 충분히 기다리는게 전제가 된 서비스라면, 조금 느린 속도도 문제 없겠지만, 화면의 콘텐츠 요소로 사용되는 추천이라든가, 사용자 행동에 반응해서 팝업을 띄워준다는가 하는 즉각적인 서비스 용도라면 단연 속도가 모델 선택의 기준이 된다.

속도는 온라인 서비스에서만 중요한 것이 아니다. 배치서비스라 할지라도 서비스로서의 머신러닝은 사용자의 반응에 즉각 대응하는 실시간성을 갖추어야 한다. 실제의 사용자들은 배치의 학습주기보다 빠르게 움직이고, 또 쉽게 싫증을 내는 존재이기 때문이다. 사용자가 모델의 최선의 결과물을 보았음에도 반응하지 않았다면, 그에 불만족한 것이므로 빠르게 콘텐츠를 전환하여야 한다. 모델의 배치 계산 주기(cycle)가 길다고 해서, 서비스가 콘텐츠를 바꾸지 않으면 사용자는 지루해 할 것이다. 또 이미 반응한 콘텐츠도 마찬가지다. 이미 본 기사나, 산 물건을 추천해 주는 것도 사용자에게 의미 없어 보인다. 때문에 사용자 경험(user experience)을 만족 시키기 위해서는 사용자 반응을 빠르게 수집하고, 새로운 콘텐츠로 즉각 대체하는 보조적 실시간 시스템이 항상 같이 구비되어야 한다.

타겟팅과 브랜딩. 부정확한 것도 때론 괜찮다.

머신러닝에서 우리가 기준점을 어떻게 설정하는가에 따라 아깝게 대상에서 벗어나는 집단이 있다. 일반적으로 이런 집단은 버려져서 사용되지 않겠지만, 실무의 세계에서는 좋은 사용처가 존재한다. 브랜딩이다.

브랜딩은 뭘까? 반복적인 시청으로 효용을 '학습'시키고 최종적으로는 효용을 '유도'하는 작업이라 생각한다. 태어날 때부터

'맛다 게보린'이라던가 '초코파이는 정(情)'을 알았던 것이 아니다. 지금 당장 아프거나, 배가 고프지 않더라도 반복적인 시청은 언젠가 필요할 때를 만났을 때, 해당 제품을 고르도록 학습시킨다. 또 브랜딩 메시지가 누적되면 없던 욕구를 유도시키기도 한다. 전혀 배가 고프지 않았는데 치킨광고를 보고 그날 저녁에 주문 전화를 한 적은 다들 있을 것이다. 타겟팅과 브랜딩은 세상을 바라보는 뷰가 약간 다르다. 타겟팅은 세상을 '정해진 것'으로 보고, 그 안에서 최적의 해를 찾는다. 브랜딩은 그에 비해 세상을 '변화시킬 수 있는 것'으로 보고 세상에 영향을 주기 위해 집행된다.

타겟팅을 아깝게 벗어난 경계 근처의 사용자 군집은 브랜딩의 대상으로 적합하다. 기준을 넘지는 못했지만 그래도 최소한의 관심이 있기 때문에 전혀 상관 없는 군집에 비해 유혹하기가 쉽다. 머신러닝은 문제에서 해답을 찾는 것이 기본 방향이지만, 나와 있는 해답 또는 오답에서 적절한 문제를 찾는 것도 일에 도움이 될 때가 많다.

딤러닝 시대에도 고전적인 머신러닝을 써봐야 한다

충분히 많은 데이터가 있다면 딤러닝은 거의 언제나 승리한다. 영상, 음성, 텍스트와 관련된 문제도 딤러닝이 월등하다. 다만 비즈니스 필드에서 자주 만나는 문제는 영상도 음성도 텍스트도 아닌 '테이블로 정리된 자료(tabular data)'인 경우가 많다. 심지어 양(volume)도 딤러닝을 돌리기엔 모자랄 때가 많다. 그럴 때는 feature 엔지니어링⁷을 충분히 거친 후, SVM(support vector machine), RandomForest, GBT(gradient boosted trees)등의 전통적인 머신러닝 방법을 써보면 기대보다 꽤 좋은 성능을 낸다.

비 딤러닝 계열의 대표주자는 gradient boosted trees 알고리즘 구현체인 xgboost다. 캐글⁸에서도 테이블화된 데이터를 다루는 문제라면, 딤러닝이 아니라 xgboost를 쓴 모델이 승자가 된 경우가 많다. 개인적 경험으로도 xgboost의 속도와 성능은 정말 '강패'에 가까워서 GPU를 써서 오랜시간 돌린 딤러닝과 CPU로 몇초만에 끝난 xgboost가 1~2%의 근소한 차이인 경우가 많았다. 사실 1~2%라면 수학적으로는 의미가 있지만, 비즈니스적인 의미 차이는 거의 없다.

클래식한 모델은 기존 시스템 인프라 활용면에서도 큰 이점을 가지고 있다. 회사의 분산처리 시스템에 태워 이래저래 편히 사용하기에는 귀한 GPU가 있어야 잘 운영되는 딤러닝 모델보다는, 정확도는 1~2% 뒤질지언정 아무 장비에서도 몇 초만에 돌아가고, 라이브러리 하나만 같이 태워서 배포해도 잘 돌아가는 고전 방법들이 아무래도 만만하다. (여러분이 머신 한대에 GPU드라이버부터 텐서플로우까지 설치하던 노고를

기억해보라. 그걸 N대에 적용해야 한다고 생각해보라. 번거로운 건 번거로운 것이다.) 그리고 실무자들과의 협업에서도 큰 도움이 되는데, 딥러닝의 경우 블랙박스 요소가 많아, 그 동작 과정을 사람이 이해하거나 재조사(inspection) 하기는 어렵다. 이에 비해 고전 방법들은 알고리즘이 명확하고, 학습된 후에 모델을 조사하여 지식을 끌어내기가 상대적으로 쉽다. 예를 들어 수없이 많은 속성중에서 추정 결과에 결정적 영향을 줬던 속성이 무엇인지(feature importance)를 찾는 일 등도 쉽게 가능하다. 이러한 발견들은 현업을 담당하는 실무자들의 지식과 결합되어 새로운 가설을 떠올리거나, 또는 마케팅등의 비즈니스 활동에 아이디어를 주는 경우가 많다. 머신만으로 모든 일이 완결 되는 것이 아닌 이상 이러한 영감(insight)들은 실제로 큰 도움이 된다.

사람의 일은 여전히 중요하다.

머신러닝을 비즈니스에 적용한다는 것은 두가지 종류의 불확실성과 싸워나가는 일이다. 하나는 데이터 부족으로 인한 불확실성이고, 다른 하나는 세상 그 자체가 가진 불확실성이다.

첫 번째부터 살펴보자. 머신러닝을 처음 공부하고 나면, 세상 모든 것을 다 예측해 낼 수 있을 것 같은 기대를 품는다. 하지만 현실 세상의 로그(log)를 보고 나면 절망에 빠지게 된다. 현실의 로그는 ‘없거나’ 또는 ‘쓸 수 없거나’ 둘 중 하나의 상태이기 때문이다. 사실 현업에서 로그는 오류를 잡기 위한 디버그(debug)용도나, 불만족 고객 응대(customer service)를 위한 내용이 주로 남겨진다. 아무래도 머신러닝을 위한 용도는 뒷전이다. 때문에 머신러닝에 쓸만한 데이터는 항상 부족하다. 동전 던져 앞면이 나올 확률을 계산한다고 해보자. 동전을 던진 횟수가 3번밖에 없다면 횟수가 너무 작아 확률을 정확히 추정하기 어렵다. 추정값을 뽑아도 믿기 어려울 것이다. 이렇게 데이터가 부족하여 발생하는 불확실성(epistemic uncertainty)이 우리가 처음 만나게 되는 불확실성이다. 그래도 이쪽은 낫다. 우리의 노력과 끈기로 시간이 지날수록 극복이 가능하다. 열심히 로그를 남기고, 좋은 모델을 쌓아올려가며 제거해 갈 수 있다.

첫 번째 불확실성을 어느정도 제거하고 나면 문제가 되는 것은 두 번째 불확실성이다. 그것은 현실 세상이 근원적으로 가지고 있는 불확실성이다(aleatoric uncertainty). 동전을 백만번 던져서 데이터가 충분히 쌓였다고 해보자. 동전의 앞면이 나올 확률이 50%로 맞출 수 있다. 여기서 추가로 동전을 100만 번 더 던진다고 동전의 앞면이 나올 확률이 증가할까? 그렇지 않다. 아무리 데이터를 많이 모았어도 앞면이 나올 확률은 항상 50%의 불확실성을 가진다. 이것이 세상 그 자체가 가지고 있는 불확실성이다.

두 번째 불확실성은 얼핏 생각하기에 조금 절망적이다. 아무리 데이터를 많이 모아도, 아무리 좋은 모델을 만들어도 극복할 수 없어보이기 때문이다. 이 때 필요한 것은 인식의 변화이다. 우리는 컴퓨터 앞에서만 일할 필요가 없다. 우리는 현실에 영향을 줄 수 있는 실제의 사람이다. 동전의 앞면이 필요하다면, 앞쪽이 잘나오도록 추를 달아도 된다. 양쪽이 앞면인 동전을 만들어도 된다.

두 번째 불확실성을 비즈니스 관점에서 생각해보자. 모든 데이터와 모델을 동원해 특정 사람이 좋아할 만한 최고의 상품을 추천할 수 있다 하더라도, 해당 소비자가 구매할 확률은 10%일 수 있다. 이때 구매율을 더 끌어 올릴 수 있는 방법은 더 좋은 수학적 추정이 아니라, 가격 할인, 묶음 상품, 할인율의 일괄 적용, VIP만을 위한 할인율 차등 적용 등 현실 세상의 행동이다. 우리의 직접적인 행동이 극복할 수 없어 보였던 두번째 불확실성을 변하게 만든다. 머신의 시대지만, 사람의 일은 여전히 중요하다. 그 점을 잊지말자.

맺음말

머신러닝은 이 시대의 꽃이다. 가장 화려하고 모두가 가지고 싶어한다. 하지만 이 꽃을 땅에 심어 자라나게 만드는 데에는 손에 흙을 묻히는 일이 필요하다. 우리가 어떤 흙들을 만져야 하는지, 때로는 어떻게 심어야 하는지에 대해 이야기했다. 모두가 아름다운 꽃을 피우기를 바란다.

^{*1} 참고 | 웹상에서 실행되며 코드와 계산 결과, 각종 차트와 그래프 등을 한 눈에 볼 수 있게 만들어진 작업 환경. 전체 과정을 한 화면에서 순서대로 볼 수 있기 때문에 작업하기에도, 타인에게 공유하기에도 편하다. 데이터 분석가들이나, 머신러닝 연구자들이 즐겨 사용한다. 주피터 환경에서 다른 언어도 사용가능하지만, 주로 파이썬 (python) 언어가 사용된다. ^{*2} 참고 | 마치 사진을 찍듯이, 특정 시기 기준의 모든 데이터를 일컫는다. ^{*3} 참고 | 님들보다 신제품등을 빨리 구입해서 사용해야 직성이 풀리는 적극적인 소비자군 ^{*4} 참고 | 모델이 데이터가 만들어진 원리를 배우기보다는 데이터 자체를 기억해버리는 경우이다. 학습시킨 데이터에 대해서는 잘 풀지만, 새롭게 보는 데이터에 대해서는 성능이 오히려 떨어진다. ^{*5} 참고 | 모델이 너무 단순하거나 또는 데이터가 부족하여, 실제 현실을 제대로 배우지 못하는 경우를 말한다. 역시나 성능이 떨어진다. ^{*6} 참고 | 보통 모델의 우위를 판단하기 위해 사용되는 A/B테스트는 여러개의 모델을 동시에 비교하게 될 경우, 어느것이 최적인지 알아내는 데 시간이 오래걸리고, 또 그 시간까지 최적이 아닌 손해보는 모델에도 사용자가 상당히 많이 노출되게 된다는 단점이 있다. 각 모델에게 동일한 사용자수를 배분하는 일반적인 A/B테스트 로직과 달리 MAB는 성능이 좋은 모델에게는 더 많은 사용자들, 성능이 낮은 모델에게는 성능을 평가할 수 있는 소량의 적절한 사용자들 노출시켜, 충분히 탐색을 하면서도 이익을 최대화 하는데 초점을 맞춘다. ^{*7} 참고 | 많은 속성(feature)들 중에서 도움이 되는 것만을 골라내거나, 여러 속성의 조합으로 새로운 속성을 만들어내는 작업을 말한다. ^{*8} 참고 | 데이터 분석, 머신러닝 전문가들이 공개된 문제에 대해 각자 더 정확도 높은 모델을 만들어내어 경쟁하는 웹사이트 http://kaggle.com

음성인식 방법과 카카오i의 음성형엔진

일반적으로 사람들이 생각하는 음성인식 서비스는 사람과 대화할 때처럼 자신의 말에 적절히 반응하거나 명령을 수행하는 등 기계가 자신의 의도를 파악해 반응하는 과정 전체를 생각한다. 이 과정은 크게 사용자의 음성을 문자 정보로 변환하는 과정, 변환된 문자 정보에서 사용자의 의도를 파악하는 과정, 적절한 액션을 취하는 과정으로 나눌 수 있다. 음성을 문자 정보로 변환하는 과정을 음성인식, 문자 정보로부터 사용자의 의도를 파악하는 과정을 자연어 처리라 부른다. 당연히 음성인식과 자연어 처리 모두 좋은 성능을 보일 때 유용한 음성인식 서비스가 가능하다. 음성인식 단계에서 틀리면 자연어 처리 성능이 아무리 뛰어나다고 하더라도 전체 시스템이 제대로 동작하지 못할 것이므로 음성인식 성능이 음성을 이용한 서비스의 전제 조건이 된다.

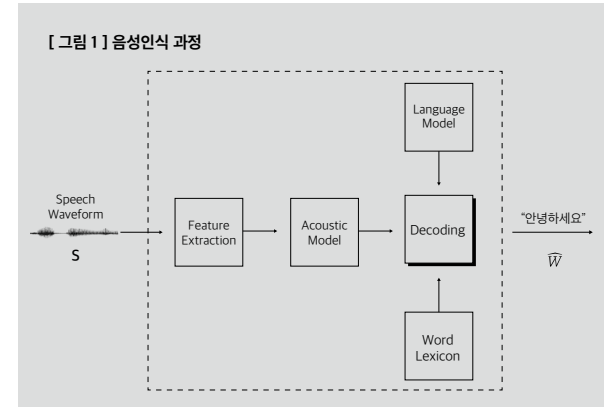
이 글에서는 자연어 처리 과정은 빼고 사람이 발성한 음성을 문자 정보로 변환하는 음성인식 방법에 대한 주요 개념을 소개하고, 카카오에서는 음성인식 성능을 높이기 위해 무슨 일을 하고 있고 이렇게 개발된 음성인식 엔진을 어떻게 활용하고 있는지에 대해 설명하고자 한다.

글 | 김훈 eldon.kim@kakaocorp.com

어쩌다 보니 이쪽 일을 20년 가까이 하고 있습니다. 이정도 했으면 이 분야에 대해 도가 터야 하는데 아직도 어렵네요. '10년 후면 사람처럼 말을 알아듣는 인식을 만들 수 있을 거'라고 생각했는데, 그 10년이 두 번째 지나고 있습니다. 다음 10년 안에는 음성인식의 모든 문제가 해결됐으면 좋겠고 그 일에 제가 조금은 기여했으면 합니다.

음성인식 개요

음성인식 과정을 단순화 하면 [그림 1]과 같이 입력된 음성에 대해 여러 단계의 처리과정을 거친 후 단어 열로 변환해 출력해 주는 것이다. 음성인식은 입력된 음성이 어떤 단어들로 이루어져 있을 확률이 가장 높은가를 찾는 문제로 다음과 같이 정의할 수 있다.



‘일정 길이 T 동안 입력된 음성 sequence X 에 대해서 인식기가 표현할 수 있는 모든 단어들의 조합 중 확률적으로 가장 가능성이 높은 단어 열 W는 무엇인가가 된다. 수식으로 표현하면 [수식 1] 과 같다.

[수식 1]

$$\bar{W}: W_1, W_2, W_3, W_4 \dots \dots, W_N \rightarrow N\text{개의 단어들로 이루어진 문장}$$

$$\vec{X}: X_1, X_2, X_3, X_4 \dots \dots, X_T \rightarrow \text{일정간격으로 추출한 T개의 음성 특징 벡터}$$

$$\operatorname{argmax}_W P(W|X) = \operatorname{argmax}_W \frac{P(X|W)P(W)}{P(X)} = \operatorname{argmax}_W P(X|W)P(W)$$

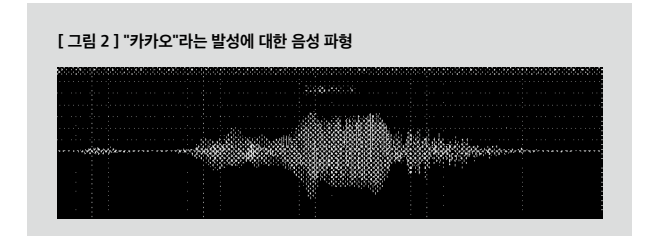
여기서 P(X|W)은 음향 모델 확률로 W라는 단어를 발성했을 때 X라는 신호적 특성이 발생될 확률이다. 사용자가 ‘카카오’ 라고 발음한다고 가정할 때 모델링 된 음향모델을 이용해서 P(X|카카오), P(X|카키색), P(X|카드) 등의 확률을 구해보면, P(X|카카오)가 가장 높은 확률 값을 보일 것이다. P(W)는 언어모델이라고 하는데 음성신호 특징에 해당하는 X가 없다. 이 값은 음성신호와 관계없이 그 음성이 무엇이라고 예측하는 값으로 어떤 단어가 말해질 확률을 미리 계산해 가지고 있다. 단어가 말해질 언어적 확률은 일반적으로 대량의 텍스트 말뭉치(text corpus)에서 단어의 발생 빈도를 계산해 구한다.

음성인식 과정은 크게 음성분석, 음향모델 계산, 언어모델 계산, 디코딩의 4단계로 나눌 수 있다.

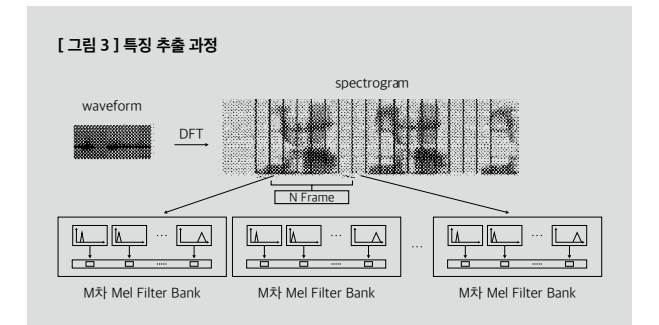
음성분석

음성신호에서 주파수 분석을 통해 음성의 특징되는 부분을 추출하는 과정을 말한다. 음성인식을 위해서는 음성을 마이크로폰(mic)을 이용해 디지털 신호로 바꾸는데 이 과정을

샘플링이라고 한다. 대체로 1/16000 초 간격으로 음성의 크기를 측정하고 각 샘플은 2byte short 로 표현한다. 이렇게 음성을 샘플링 했을 때, 1초의 음성이 가질 수 있는 경우의 수는 $2^{16000 \times 2 \times 8}$ 이다. 이 값은 대략 10^{100000} 정도로 1 다음에 0을 10만개 써야 할 만큼 큰 수이다. 이는 같은 사람이 같은 단어를 여러 번 발음한다고 해도 동일한 값이 나올 확률은 0에 가깝다는 뜻이다. 이렇게 항상 다른 값을 갖는 데이터 이지만 사람은 어떤 음성이 ‘가’인지 ‘나’인지 누구의 소리인지 등을 구별하며 의사 소통을 한다. 이렇게 할 수 있는 이유는 최소한 사람이 음성을 인식하는 과정에서 음성을 샘플 단위 하나 하나로 구별하며 인식하지 않기 때문일 것이다. 음성이 가질 수 있는 다양한 변이를 훨씬 작은 차원으로 줄인 후 어떤 특징을 인지하기 때문에 말을 알아들을 수 있다.



위에 보이는 그림은 ‘카카오’ 라고 발성했을 때의 음성파형이다. 발성 기관인 성대의 떨림 정도와 성도의 모양 변화에 따른 공기 흐름으로 여러 소리가 만들어 지는데 성대의 떨림 정도나 모양이 급격하게 변할 수 없으므로 음성은 짧은 구간(보통 0.02초)동안은 안정적이라 가정하고(quasi-stationary), 이 단위로 음성을 분석하며 소리가 만들어진 상태를 예측한다. 0.02초 길이의 음성 파형을 어떤 주파수적인 특성을 갖는지 분석하는 등 여러 단계의 신호처리 과정을 거쳐 최종적으로 수십 개의 숫자들로 표현하는데, 이 값들을 특징 벡터라고 한다.



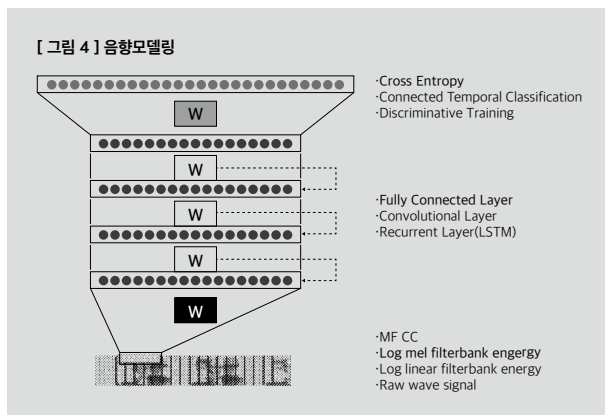
음향모델

음향모델링은 음성을 0.02초 구간을 0.01초씩 시간축에 따라 움직이며 만든 특징 벡터열 \vec{X} 와 어휘 셋 W에 대해 P(X|W) 확률을 학습하는 과정이다. 고전적인 음성인식은 음소를 GMM(Gaussian

Mixture Model)으로 모델링하고 이 음소들의 연속적 변화를 HMM(Hidden Markov Model)으로 예측하는 GMM-HMM 방식인데 GMM 확률 모델 부분만을 딥러닝(deep learning, DNN)으로 대체하는 방법(DNN-HMM¹⁾) 과 HMM으로 음성의 변화를 예측하는 부분까지 포함하여 신경망으로 대체한 종단 간(end-to-end) 방식으로 음성인식을 구현하는 많은 시도가 있다^{2,3)}. 종단 간 방식은 기존 음성인식 구현을 위해 필요한 신호처리, 발음변환, 언어모델, 디코딩 단계에 전문적인 지식이 개입하는 것을 최소화하면서 이 부분의 모델링을 신경망이 학습하게 함으로써 뛰어난 성능을 보이고 있다. 하지만 아직은 GMM 확률 모델링 부분만 DNN으로 대체한 DNN-HMM에 기반한 음성인식이 가장 좋은 성능을 보인다.

End-to-End 기반 음성인식이 번역에서처럼 기존 통계적 방법을 뛰어넘지 못하는 것은 입력인 음성 sequence 길이 T 가 출력인 문자열 개수 N에 비해 아주 긴 프로세스이기 때문이 아닐까 한다. 1초짜리 음성을 샘플로 보면 T=16,000, 특징벡터를 추출하는 간격으로 봐도 T=100 sequence이고 이때 output sequence N=1~3개로 T>>N 인 프로세스이다.

DNN으로 모델링 할 때 신경망(neural network) 구조를 CNN(convolutional neural network), LSTM(long short-term model), FNN(fully connected neural network) 중 어떤 구조를 사용할 것인지에 따라 약간씩 성능 차이는 있지만 그 차이는 그리 크지 않다. 단어 W는 음소 단위 조합으로 만들어졌다고 보고 일정 간격으로 발생하는 특징 벡터에 대해 이 시점 어떤 음소가 가장 높은 확률을 가져야 하는지를 타겟(target)으로 모델을 학습한다. [그림4]은 DNN 음향모델을 표현한 것이다.



음향모델 학습을 위해서는 음성데이터와 이 음성데이터는 무슨 문장을 발성한 것인지에 대한 원고(script)가 있어야 한다. 음성데이터 수집을 위해 원고를 주고 사람들에게 읽게 해서 녹음하거나 음성만 녹음된 데이터가 있을 경우 이를 사람이 듣고 정답을 달아주는 옴겨쓰기(transcription) 작업이 필요하다. 원고를

주고 읽어서 수집한 데이터는 아무래도 실제 사용자 환경을 반영했다고 볼 수 없다. 그러므로 이 데이터를 이용해 프로토타입을 만들고 그 후 서비스를 하면서 입력되는 로그 데이터를 활용하여 실제 사용자 환경을 반영한 음향모델을 학습시킨다.

언어모델

언어모델은 방대한 텍스트를 분석해 모델을 만들어 현재 인식되고 있는 단어들 간의 결합확률을 예측하는 과정이다. 언어모델에서는 특정 단어 다음에 나올 단어의 확률 추정이 이뤄진다. 언어모델은 단어들 간의 관계를 확률로 나타내어 서로 관계가 높은 단어들이 결합할 가능성을 높이는 역할을 한다. 언어모델을 이용하면, 발음이 비슷한 단어가 많거나 불분명하게 발음되어 음향모델로만 판단했을 때 잘못된 인식이 발생할 위험을 낮춘다. 예를 들어 '나는 점심으로 밥을 먹었다' 라고 발음했다고 할 때 '점심으로' 다음에 음향모델에만 의존하는 '밥'이 발음이 비슷한 '방', '밤' 등으로 잘못 인식될 수 있다. 언어모델은 이러한 문제의 발생을 차단하는 역할을 하는 것이다. 언어모델은 탐색할 필요가 없는 단어를 사전에 걸러 내어, 결과적으로 음성 인식 속도를 높이는 데 기여한다.

N개의 단어로 구성된 문장 W W₁, W₂, W₃ ..., W_N에 대해서 문장 생성 확률은 다음과 같다.

$$P(W) = \prod_{k=1}^N P(W_k | W_{k-1} W_{k-2} \dots W_1)$$

이 때 문장이 길어짐에 따라 확률 계산이 현실적이지 않게 되어 최근 n개의 history로 근사화 한다. 이를 n-gram 모델이라고 하는데 대용량 데이터를 이용하여 통계적 모델로 학습한다. N-gram 모델링 부분을 DNN 기반으로 학습한 많은 시도가 있었고 높은 성능을 보이는 경우도 있지만 음성인식에서는 아직 통계적 n-gram 방법에 비해 비효율적이라 많이 사용되지 않는다.

언어모델은 해당 도메인의 텍스트(text) 데이터를 최대한 많이 수집한 후 단어들 간의 통계적 특성을 추출한다. 이때 서비스 하고자 하는 도메인에 적합한 데이터를 최대한 많이 수집해야 하는 것은 기본이며 수집된 데이터에 포함된 오타나 띄어쓰기 오류 등을 얼마나 잘 수정했는지에 따라 언어모델의 성능이 좌우된다.

디코딩

음향모델과 언어모델로 구성된 탐색 공간에서 가장 최적인 경로를 찾는 과정으로 음성이 어떤 단어열을 나타내는지를 추정한다. 연속음성인식 디코딩은 문장을 이룰 단어의 개수와 각 단어의 경계에 제한을 두지 않고 탐색하는 것으로 그 경우의 수가 크다.

인식해야 할 음성이 N 개의 단어라고 가정하고 음성의 발화 길이가 T개라고 하면 이는 T 개의 frame에서 N-개의 경계를 찾는 문제로 발생할 총 가지 수는 $T^C N - 1$ 이다. 5초의 음성이 6단어 정도로 이루어 졌다고 만 해도 T=500, N=6 이므로 $500^6 5 =$ 약 2500억개 경우를 탐색해야 한다. 이 탐색과정을 효율적으로 수행하기 위해 음성인식에서는 크게 Lexical Tree 기반 디코딩과 wFST(Weighted Finite State Transducer) 기반 디코딩(decoding)을 이용한다.

Lexical Tree 기반 디코딩은 HMM, Context 정보, 발음열, 문법정보를 별도로 구현한 후 인식 시 네트워크를 만들어 가면서 탐색한다. Lexical Tree는 음향학적 특성이 같은 노드들을 공유해 탐색공간을 줄일 수 있지만 leaf node에 도달하기 전에는 언어모델을 적용할 수 없어 인식률이 저하될 수 있다. wFST는 HMM 모델, context 정보, 문법정보에 사용되는 망을 하나의 통일된 틀로서 결합하고 최적화할 수 있는 장점이 있다. 최근에 발표되는 대어휘(large vocabulary) 음성인식 엔진 대부분이 wFST 기반의 디코더 기술을 사용하고 있는 것으로 알려져 있다⁴⁾. wFST는 결합한 네트워크를 미리 만들어 놓고 탐색하는 방식으로 빠르고 정확하지만 어휘나 언어모델이 약간만 변경되어도 네트워크를 다시 구성해야 하고 어휘와 언어모델이 커지면 네트워크 수립 시에 많은 메모리가 필요하다.

카카오 음성인식/합성 기술의 API 공개

음성인식을 이용하면 기기의 복잡한 기능을 사람과 대화하듯이 자연스럽게 풀어주고 단계를 거쳐야 하는 기능을 한번에 쉽게 사용할 수 있는 등 인터페이스로서 많은 장점을 가지고 있어 오래전부터 지속적인 연구개발이 진행되어 온 분야이다. 오랜 역사를 가지고 여러 시행착오를 거치며 음성인식 기술 각 단계별로 특징 벡터는 무엇을 사용해야 하고 음향모델링은 어떻게 하는게 좋은지에 대한 표준이 거의 정해져 있다. 최근 음성인식기의 성능이 많이 향상된 것은 딥러닝을 이용한 음향모델링의 개선을 제외하면 특별한 알고리즘이 새로 개발되었기 때문이 아니다. 카카오 음성기술도 다른 음성인식 업체와 비교해 알고리즘이나 기술에 있어 큰 차별점은 없다. 단지 실수 없이 기본에 충실하게 개발된 엔진이 있고 이를 적용할 서비스를 운영하고 있으며 이 과정에서 축적되는 데이터를 잘 활용하고 있을 뿐이다.

음성인식 개발 과정을 보면 결국은 기계를 학습시키는 것으로 다양한 데이터를 많이 학습할 수록 뛰어난 성능을 발휘한다. 일정 수준 이상 되는 프로토타입 인식기를 개발하고 이를 적절한 서비스에 적용하면 사용자로부터 실제 상황의 데이터가 수집되고 이를 이용해 성능을 개선하면 편리성을 느낀 사용자가 더 많이 사용하면서 더

다양한 데이터가 축적되고 이는 다시 성능 개선에 활용된다.

카카오는 개발된 음성인식/합성 기술을 카카오 자체 서비스인 다음, 카카오톡, 카카오톡, 카카오택시, 카카오프라이버, 카카오톡 치즈⁵⁾, 멜론⁶⁾ 앱 등 다양한 분야에 적용하고 있다. 대부분 음성인식 기술을 이용한 음성검색이며 음성합성은 다음 뉴스를 음성으로 읽어 주는 서비스에 활용되고 있다. 이렇게 서비스를 운영하며 쌓인 데이터로 업그레이드 된 음성인식 엔진은 다른 회사에서도 성능을 인정받아 자체 서비스 이외에 현대차⁷⁾, LG유플러스⁸⁾, KT⁹⁾ 에도 음성기술을 제공하고 있다. 카카오에서 개발한 음성인식/합성 엔진은 기술을 공유해 많은 영역에서 음성을 기반으로 한 발전을 가능하게 할 목적으로 오픈API로 공개하였다. 2014년 2월 음성인식, 2014년 6월 음성합성 기술을 하루 5000회 호출까지 무료로 사용할 수 있는 형태로 오픈했고 2017년 1월부터 무료이용 기준 범위를 하루 2만 회까지 확대했다. 현재까지 카카오 음성 API를 이용해 음성관련 서비스를 하고 있거나 시도해 본 기업 및 개인은 470여 곳이다. 쇼핑, 교육, 병원 등 다양하게 분포돼 있다. 특히 주소나 지역명 검색 빈도가 높은 업종에서 효율적으로 사용되고 있다. 카카오 음성인식 API는 다음과 같은 4가지 서비스 모드를 지원한다.

[카카오 음성인식 API 4가지 서비스]	
· 고립어(service type word)	· 연속어(service type dictation)
· 웹검색(service type web)	· 지도(service type local)

카카오가 제공하는 음성 API 이용에 대한 자세한 내용은 카카오 디벨로퍼스 홈페이지(<https://developers.kakao.com/features/platform#음성>)을 참조하면 된다.



카카오 음성인식/합성 기술의 API 공개

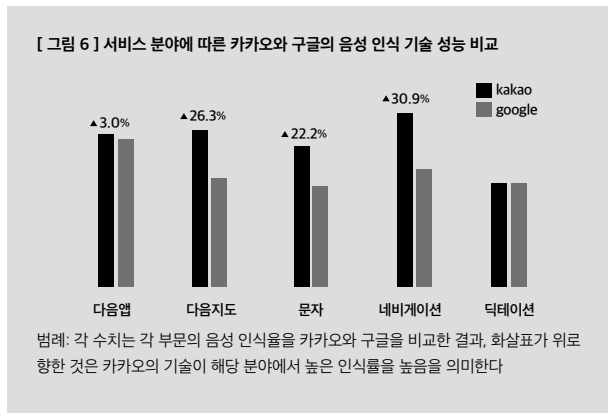
음성인식은 인식방법에 따라 고립어 인식과 연속어 인식으로 나눌 수 있다. 고립어 인식이란 정의된 어휘 중 어느 한 단어만을 발음

했을 것으로 가정하고 인식하는 것이며, 연속어 인식이란 정해진 어휘들의 조합으로 발생할 수 있는 모든 경우를 발음했을 것으로 가정하고 인식하는 것이다.

예를 들어, 음성인식기가 인식할 수 있는 단어가 ‘카카오’, ‘인공지능’ 두 단어라고 하면 고립어 인식은 입력된 음성이 ‘카카오’나 ‘인공지능’ 둘 중 하나일 것으로 가정하고 ‘카카오’ 아니면 ‘인공지능’ 이라고 답한다. 연속어 인식은 ‘카카오 인공지능’, ‘카카오 인공지능 카카오’, ‘카카오 카카오 카카오’ 등과 같이 두 단어의 조합으로 만들어질 수 있는 어떤 어휘도 모두 인식하는 방법이다. 검색어, 주소, 지역 정보(point of interest, POI) 인식과 같이 인식해야 할 어휘의 조합이 거의 무한대인 경우는 연속어 인식 기술을 사용해야 한다. 고립어 인식은 명령어나 ‘예’, ‘아니오’ 와 같이 사용자의 발화 형태를 명확히 예측 할 수 있는 상황에서 사용하면 연속어 방식에 비해 높은 인식 성능을 얻을 수 있다. 고립어 모드에 사용할 어휘는 개발자가 텍스트 형태로 정의해서 입력해 주면 그 어휘 중에서만 인식한다. 음성 검색, 로컬 검색, 딕테이션에 적용된 연속 음성인식기의 인식사전은 100만 단어 규모로 100만 단어들의 조합으로 가능한 모든 문장을 인식대상으로 한다.

웹검색이나 지도검색 모두 기본적으로는 연속어 인식 방법이지만 언어모델을 그 검색이나 로컬 분야에 대해 특화시켜 놓은 것만 다르다. 음성인식기를 어떤 분야에서 주로 사용할 것인지 예측할 수 있다면 그에 적합한 서비스 모드를 선택해 일반 연속어 모드를 사용하는 것보다 높은 인식률을 얻을 수 있다. 예를 들어 지도 검색의 경우는 인식해야 할 POI는 수백만 개 이상 이지만 지도 검색에서 발생할 만한 패턴을 잘 모델링 할 수 있으므로 다른 서비스 도메인에 비해 인식률이 높다.

[그림 6] 은 언어모델을 세분화 해 잘 모델링하면 하나의 공통된 언어모델을 사용하는 것으로 보이는 구글 음성인식 API보다 얼마나 좋을 수 있는지 실험한 결과이다.



두 엔진은 일반적인 문장에 대해 테스트한 딕테이션(dictation) 영역에서는 차이가 없지만 지도검색 분야에서는 지도검색에서

사용할 만한 문장들로 최적화된 언어모델을 사용하는 카카오의 음성형 엔진이 구글보다 30% 정도 높은 인식률을 보인다. 딕테이션은 모든 음성인식기가 사람말을 받아 쓰는 것을 의미하는데, [그림 6]에서 딕테이션은 영역 제한을 두지 않고 랜덤하게 문장을 추출하는 것을 뜻한다. 이와 같이 음성인식을 사용할 범위(domain)를 적절하게 한정하면 인식률을 비약적으로 높일 수 있다. 카카오의 음성 인식 기술 대상은 현재까지는 한국어에 한정되어 있다.

마치며

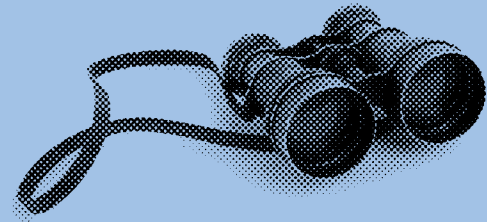
최근 음성인식 성능이 비약적으로 향상 되었지만 아직 사용자가 원하는 수준에 미치지 못하는 부분이 많다. 특히 잡음, 반향, 사용하는 디바이스 특징 등으로 인해 음성신호가 학습데이터와 일치하지 않는 상황이나 간투사가 빈번이 발생하고 한번에 발음하지 못하고 버벅거리며 일반적인 어순과 다르게 발음하는 경우가 많은 일반 대화체 상황에서 성능이 급격하게 하락한다. 음성인식에 사용되는 모델링 방법이 사람이 음성을 이해하는 방법을 잘 모델링 했다면 데이터가 많이 투입되고 컴퓨터의 계산 능력이 발전한다면 현재까지 정립된 방법론을 이용해 언젠가는 사람과 비슷하거나 뛰어넘는 음성인식기가 나올 것이다. 하지만 방법이 잘못 되었거나 음성이해 과정을 제대로 모델링하지 못한 부분이 있다면 데이터만 늘린다고 해결 되지 않을 것이다. 카카오는 기존 방법으로 양질의 데이터를 잘 활용해 학습하는 방법과 새로운 접근법을 찾는 연구를 병행하며 인식성능을 개선시키기 위해 노력하고 있다.

***1** 참고 | G. Hinton, Deep Neural Networks for Acoustic Modeling in Speech Recognition, IEEE Signal Processing Magazine, 2012. ***2** 참고 | Y. Zhang, Towards End-to-End Speech Recognition with Deep Convolutional Networks, arXiv:1701.02720, 2017. ***3** 참고 | D. Amodei, End-to-End Speech Recognition in english and mandarin, ICLR 2016, 2016. ***4** 참고 | 권오욱, WFST 기반 음성인식 연구동향, 제 21권 제 2호, 컴퓨터정보통신연구, 2013. ***5** 참고 | <http://v.media.daum.net/v/20170727111503886> - 치즈앱에 음성인식 적용 관련 기사. ***6** 참고 | <http://v.media.daum.net/v/20170811084822273> - 멜론앱에 음성으로 음악검색 적용 관련기사. ***7** 참고 | <http://v.media.daum.net/v/20170725030405241> - 현대차 관련 기사. ***8** 참고 | <http://v.media.daum.net/v/20140310151110273> - LG유플러스 관련 기사 ***9** 참고 | <http://v.media.daum.net/v/20130912142710116> - KT 관련 기사

생생한

AI 현장의

이야기



“눈으로 보았는가? 그렇지 않다면 되돌아 가라.” 어느 한 부대 지휘관 사무실의 문 앞에 써져 있던 글귀입니다. 현장 확인의 중요성을 강조한 말입니다. 책으로 배우는 지식도 값지지만, 현장과 괴리된 지식은 현실과 관련된 논의에서 공허한 제안일 뿐입니다. AI 시대는 세미나와 포럼의 시대로도 볼 수 있습니다. 갖가지 주제로 열리는 AI 행사를 일일이 다 열거하기도 어루울 정도입니다. 그래도 현장에 가서 듣고, 현장의 청중 그리고 연사들과 이야기하는 것은 또 다른 관점의 문을 열게 하는 시발점이 되곤 합니다. 그래서 이번 호를 시작으로, 주요 학회 관람기를 카카오 AI 리포트에서는 소개하려 합니다. 그 첫 번째로 충남대 로스쿨의 이상용 교수님이 런던에서 열린 국제인공지능법학회를 참관하신 참관기를 담았습니다. 그 다음 글은 AI가 접목된 의료영상 업계의 양상을 현장에서 활약하고 있는 주요 기업들의 이야기를 중심으로 살펴보았습니다.

국제 인공지능 법학회 참관기

언제쯤이었을까. 아마 이율배반의 재귀적 순환에 관한 더글러스 호프스태터(Douglas Hofstadter)의 명저 “괴델, 에셔, 바흐”¹에 매료되었을 때부터였지 않았나 싶다. 그 후 한스 모라벡(Hans Moravec)², 레이 커즈와일(Ray Kurzweil)³, 닉 보스트롬(Nick Bostrom)⁴ 등의 저작을 읽어 가면서 나는 점점 인공지능의 세계에 빠져들었고, 현재의 기술 수준이 아직 걸음마 단계에 있음을 알게 된 뒤에도 관심은 커져만 갔다. 온라인 그룹⁵을 만들어 별로 내키지 않아하던 동료 판사들을 억지로 가입시키기도 했었는데, 생각해보면 2015년 법원을 그만두고 로스쿨로 옮기게 것도 인공지능과 법이라는 분야에 좀더 시간을 쏟고 싶었던 마음이 컸던 것 같다.

런던으로

동료 교수들과 이 주제에 관하여 토론하는 것은 매우 즐거운 일이었다. 우리는 곧 인공지능과 법을 공부하는 모임을 만들었는데, 이 모임은 이제 ‘한국인공지능법학회’라는 어엿한 이름을 갖게 되었다. 그리고 올해 6월, 국제인공지능법학회(International Association for Artificial Intelligence and Law, IAAIL)에서 격년으로 개최하는 16번째 컨퍼런스가 런던에서 열렸다.

언어은 테러 보도에 걱정이 되기도 했지만, 우리는 잔뜩 보험에 가입한 것을 위안으로 삼아 런던으로 출발했다. 6월 런던의 날씨는 좋았고 컨퍼런스가 열리는 킹스칼리지는 200년에 가까운 역사를 드러내듯 고풍스럽고 고즈넉했다. 근처에는 빅토리안 고딕 스타일로 지어진 로열코트(Royal Courts of Justice)⁶와 우리나라의 사법 연수원 격으로 법정 변호사(barrister)를 키워내는 4개의 인(inn)이 있었다. 이들을 둘러싸고 법조타운이 형성되어 있는데, 골목마다 들어찬 법률사무소들 사이로 정장을 입은 채 기록 가방을 바삐 끌고 있는 젊은 변호사들이 인상적인 곳이었다.

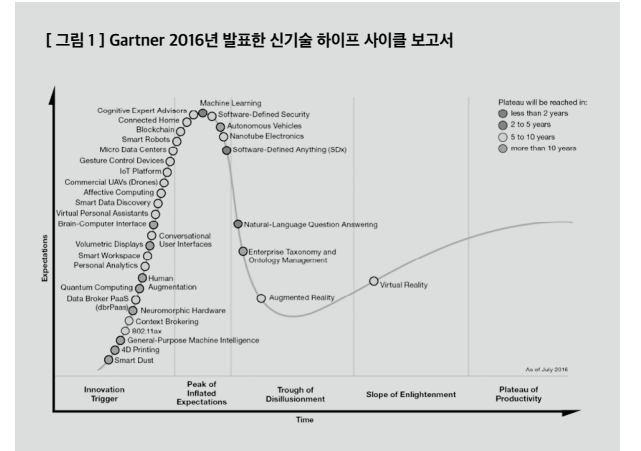
기본 좋은 기대와 흥분으로 강당에 자리한 나는 곧 이번 컨퍼런스 일정이 만만치 않을 것임을 깨달았다. 컴퓨터공학을 중심으로 여러 학문 분야에서 심도 있게 이루어지는 발표 내용을 이해하는 것은 법학을 전공한 필자로서는 벅찬 일인 것만 같았다. 그러나 힘겹게 발표 내용들을 따라가면서 적어도 인공지능법 분야의 범위와 주요 주제, 그리고 최신의 트렌드를 어렵풋이나마 스케치할 수는 있겠다는 생각이 들었다. 이 기고문은 그러한 고민의 결과물이다.

국제인공지능법학회

‘인공지능과 법(AI and Law)’ 또는 ‘법률 인공지능(Legal AI)’라는 분야는 생각보다 오랜 역사를 갖고 있다. 유럽에서는 1980년대 후반에 이미 연구자 커뮤니티가 형성되었고 1990년대를 지나면서는 확립된 연구분과로 인정받게 되었다. 이러한 학문적 배경을 바탕으로 1992년 공식적으로 창립된 국제인공지능법학회는 ‘인공지능과 법(Artificial Intelligence and Law)’이라는 권위있는 잡지를 발행하고, 격년으로 컨퍼런스(International Conferece on Artificial Intelligence and Law, ICAIL)를 개최하는 등 이 분야의 선도적 역할을 맡아 왔다.

2010년대에는 보다 상업적 측면에 초점을 맞춘 리걸 테크(Legal Tech) 영역이 출현하면서 법률 인공지능 기술을 바탕으로 실용화된 법률 서비스도 많이 생겨났다. 미국의 Ross Intelligence는 IBM의 인공지능 플랫폼인 왓슨(Watson) 기술을 활용하여 자연어 법률정보 검색 서비스를 제공하고 있고, 카이라(kira)는

법률문서로부터 정보를 추출하여 데이터베이스로 만드는 서비스를 제공하고 있다. 그 외에도 수백 개의 기업이 마케팅, 문서 자동화, 업무관리, 법률정보 검색, 법률교육, 온라인 분쟁해결, e-디스커버리, 데이터 분석, 컴플라이언스 등 여러 영역의 법률 서비스를 제공하고 있다⁷. 법률 인공지능 기술에 대한 대중의 기대에는 상당한 거품이 끼어 있는 것으로 보이지만[그림 1], 그림에도 불구하고 이 분야의 가능성은 그 어느 때보다도 넓게 열려 있다.



이러한 분위기에서 개최된 이번 컨퍼런스는 예년보다 백 명 가까이 많은 280여 명이나 참석하였고, 5일 동안 10개나 되는 워크샵이 함께 열리면서 백 건이 넘는 발표가 이어지는 등 그 어느 때보다도 많은 관심 속에 치러졌다⁸. 규모도 컸지만 이번 컨퍼런스에서 보다 주목해야 할 부분은 외연의 확대였다.

컴퓨터과학자와 기업

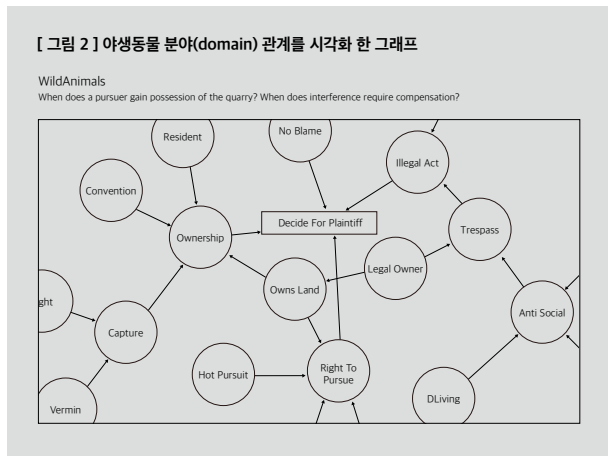
컴퓨터과학자들의 커뮤니티에서 출발한 법률 인공지능 분야의 발전은 연구자들의 학문적 열정과 기업의 실용적 요구에 의해 이루어져 왔다. 이들 둘은 항상 일치하지는 않았다. 2013년 국제인공지능법학회의 회장을 맡았던 윈켈스(Radboud Winkels)는 이런 고민을 “업계에서 요구되는 종류의 연구는 우리의 연구 열정과는 부정적인 상관관계에 있는 것처럼 보인다”고 익살스럽게 표현하기도 했다⁹.

그러나 이론과 현실 사이의 이러한 긴장 관계는 결과적으로 서로 상승 작용을 일으켜 좋은 성과를 낳는 배경이 되었다. 그 좋은 사례는 법률 분야에서의 컴퓨터 논증(computational argumentation) 연구이다. 이 영역은 2003년 Katie Greenwood와 Trevor Bench-Capon 등의 연구¹⁰에서 비롯되었다. 사례 기반 추론(case-based reasoning), 비형식 논리학(informal logic), 컴퓨터 논증 모델 등에 관한 기존의 지식을 활용하여 이루어진 위 연구는 다분히 이론적인 것이었다. 그러나 연구자들은

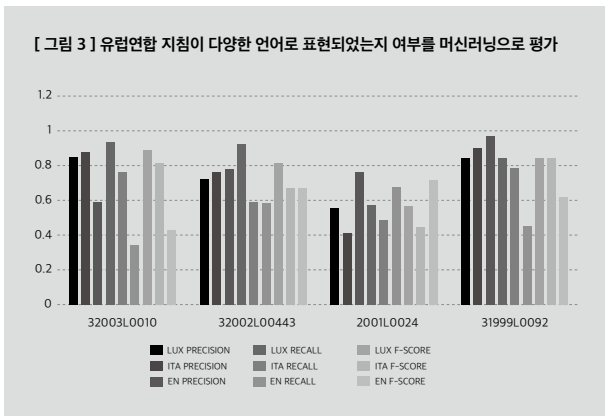
글 | 이상용 colinlee1973@gmail.com

인공지능의 매혹은 이내 사람이란 무엇인가라는 의문으로 이어졌다. 아리스토텔레스 이후로 갈라지기만 했던 질문들이 다시 한 데 모이는 듯했다. 처음부터 그게 궁금했다. 진실은 무엇일까.

곧 수요자의 요구를 반영하여 이론적 배경을 알지 못하는 사용자들도 쉽게 활용할 수 있는 전자정부(e-government) 툴을 개발하였다(Parmenides Tool)¹¹. 그 뒤에도 학계의 흐름은 추상적인 이론 연구(argumentation framework)¹²와 실용적 적용(Integrated Method for Policy making using Argument modelling and Computer assisted Text analysis, IMPACT)¹³ 사이를 오가며 바뀌어왔다. 최근에는 법적 추론의 변증법적 구조를 담아내기 위한 이론적 틀(Abstract Dialectical Frameworks for legal reasoning, ADF)¹⁴이 제안되었고, 그 후 업계의 요구에 따라 이러한 이론적 성과를 실용화하려는 작업(ADF for kNowledGe Encapsulation of Legal Information from Cases, ANGELIC. [그림 2])¹⁵이 진행 중에 있다. 그 목표는 어떤 법률 분야의 지식을 ADF 형태로 포착하여 실제 사안의 판단을 위하여 활용할 수 있는 도구를 개발하는 것이다.



법률 인공지능 기술의 주된 수요자는 로펌이나 법률정보업체 등 법률서비스 기업이었지만 그 범위는 점차 확대되고 있는 모습이다. 최근 들어서는 컴플라이언스 등 일반 기업의 법무 조직은 물론, IMPACT 사업의 경우에서 보듯 전자정부 또는 전자민주주의 아젠다에 따라 정부나 공공기관 역시 중요한 수요자가 되어 있다. 법학 연구자들이 수요자로 등장하고 있는 점도 주목할 만하다. 예를 들어 이번 컨퍼런스에서는 유럽 연합 지침(directive)이 회원국의 국내법에 수용되어 있는지 여부를 자동적으로 탐지해내기 위한 모델(Unifying Similarity Measure, USM. [그림 3])이 발표되었다¹⁶. 비록 각국의 법체계를 고려하지 않은 텍스트 비교라는 한계가 지적되기는 했지만, 장차 비교법 연구를 비롯하여 법학 또는 정책 연구자들에게 실질적인 도움을 줄 수 있는 잠재력을 지닌 연구라고 생각된다.



컴퓨터과학자와 기업

법률 인공지능 분야는 ‘법률’에 관한 인공지능 연구를 표방하였음에도 불구하고 실제 법률가들과의 교류는 모델이나 시스템 개발을 위한 소송문서 데이터의 확보를 중심으로 피상적으로 이루어져 왔다(AI for law). 그런데 최근 들어 전반적인 인공지능 기술의 발전으로 인해 생산성이 비약적으로 향상되고 사회경제에 커다란 영향을 미치게 됨에 따라 그러한 현상을 어떻게 법적·제도적으로 수용할 것인가 문제가, 법률 전문가나 정책 전문가들이 인공지능에 관심을 갖게 되었다(law for AI).

이러한 흐름을 반영하여 국제인공지능법학회는 법률 전문가들과 법률 인공지능 연구자 사이에 다리를 놓아주기 위해 이번 컨퍼런스에서 처음으로 ‘인공지능과 법실무 워크숍(1st AI and Legal Practice Workshop)’을 개최하였다. 이 취지는 큰 호응을 얻어 전체 참석자 280여 명 가운데 95명이나 이 워크숍에 참석하였고, 그 중 상당수가 법률가들이었다. 주최 측은 컴퓨터과학에 관한 지식이 부족한 이들을 위하여 법률 인공지능의 기초적 지식에 관해 소개하는 강좌를 마련했고, 이어서 이루어진 워크숍의 주제들도 지적재산권 문제나 공유 데이터셋의 부족 등 법적·정책적 쟁점이 주가 되었다. 또한 최근 많이 이슈가 되는 블록체인(block chain)과 스마트계약(smart contract)에 관하여 별도의 워크숍이 마련되었고¹⁷, 법적 판단과 의사결정을 다루는 법정심리학 분야의 워크숍도 함께 개최되었다¹⁸. 본회의 격인 ICAIL에서도 인공 에이전트(artificial agent)의 법인격 인정 여부의 문제¹⁹, 자율주행차의 규범 준수 문제²⁰, 유럽 일반 데이터보호 법제(GDPR : General Data Protection Regulation) 하에서의 법률 인공지능 활용시 법적 문제²¹, 법적 판단에 있어서 인공지능 기술 활용의 증대가 법의 지배(rule of law)라는 헌법적 원리와 상충하는지 여부에 관한 문제²², 법경제학이 판사들, 나아가 법리에 미치는 영향을 기계학습 기술을 이용하여 분석하려는

접근법²³ 등 전통적인 법률 인공지능의 영역을 넘는 주제에 관한 발표가 이루어졌다. 특히 인공지능과 법의 상호관계가 가장 첨예하게 나타나는 자율주행차와 관련하여, 전통적인 법률 인공지능 연구자들이 이를 자신들이 활약할 새로운 영역으로 인식하면서 윤리적 기준의 설정과 유지 방안의 제시에 자신감을 내비치는 모습이 인상적이었다²⁴. 다만 법률 인공지능 연구자들은 인공지능에 법인격을 부여하려는 입장에 대하여는 대체로 거부감을 보였다. 법률가에게 쓸만한 도구를 마련해주는 것조차 얼마나 어려운 일인지를 수십년간 직접 느껴왔던 이들로서는 당연한 일인지도 모른다.

일부 열정적인 법률가나 컴퓨터과학자는 상호 교류를 넘어서 상대방 영역의 본령에 뛰어들기도 했다. 일종의 법률 인공지능 경연대회인 COLIEE에 참여한 일본의 원로 컴퓨터과학자가 변호사시험에 도전했다는 소식은 컨퍼런스 내내 화제가 되었는데, 안타깝게도 시험에 합격하지는 못했다고 한다. 전통적인 법학 교육을 받은 법률가나 변호사가 컴퓨터과학을 공부하여 훌륭한 발표를 한 경우도 많았다. 특히 통구권을 포함한 여러 지역의 로스쿨에서 공부하는 박사과정 학생들이 법률 인공지능 분야에 관한 논문들을 발표하는 모습이 인상적이었다. 종래 미국 스탠포드 로스쿨은 CodeX²⁵라는 프로그램을 마련하여 법률정보학(Legal Informatics) 분야를 선도하여 왔는데, 법률 인공지능 분야의 연구에 로스쿨이 적극적으로 참여하는 흐름은 이제 세계적으로 확대되는 모습이었다.

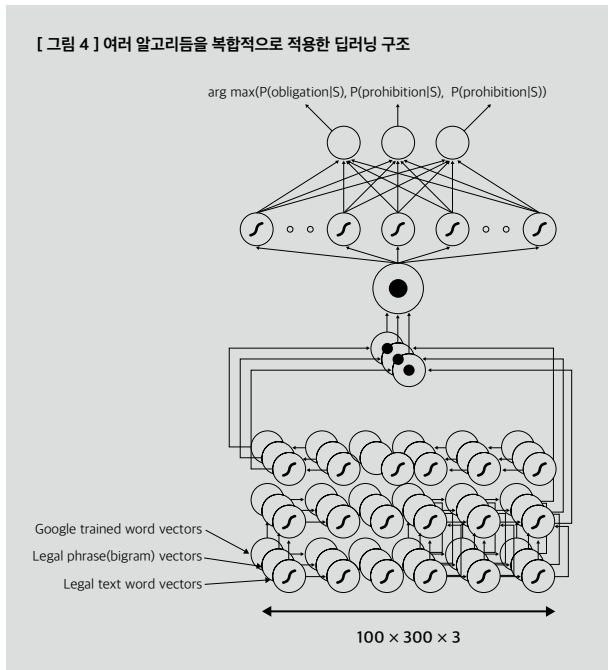
새로운 트렌드

이번 컨퍼런스에서는 사안의 파악, 법률 정보의 검색과 학습, 결론의 도출, 결론을 정당화하는 논증 등 법실무의 모든 과정에서 유용한 도구를 제공하기 위한 수많은 이론적, 실용적 시도들이 발표되었다.

특히 법적 논증 분야에서 여러 가지 흥미로운 시도가 있었다. 법률 분야에서 기존에 많이 활용되던 규범 기반 의미론적 접근방식(norm-based semantics)²⁶을 벗어나 가능세계(possible worlds) 개념을 활용한 양상논리(様相論理, modal logic)의 일종인 규범논리(deontic logic) 구조를 활용하는 시도가 대표적이다²⁷. 그 외에도, 규범 체계의 실제 구조를 경험적 측면에서 파악하려 하거나²⁸, 시간적 요소를 포용하려 하거나²⁹, 확률적 요소를 내포하는 퍼지 논리(fuzzy logic)를 활용하여 목적론적 해석을 표현하려 하거나³⁰, 심지어 어떤 규정에 대한 여러 해석 가운데 가장 강력한 것을 골라내는 논리를 구축하려 시도한 경우도 있었다³¹.

-> 전에는 단어의 의미가 어떻고, 이렇게 단어 해석에 집중했다면, 규범의 set를 상정해놓고, 가장 타당한 걸 선택하는 방식..

그러나 역시 가장 두드러지는 트렌드는 기계학습, 특히 딥러닝 기술의 적극적 활용이었다. 본 컨퍼런스에서 발표된 많은 연구, 특히 예측(prediction)을 주제로 하는 연구에서 기계학습 기술이 중요한 도구로 활용되었다. 법해석상의 모호성 문제를 해결하기 위한 방편으로서 규범논리(deontic logic)에 의한 접근방법의 한계를 지적하면서, 확률의 형태로 문맥(context)을 포착하고 CNN(convolutional neural network), RNN(recurrent neural network), Bi-LSTM(long short-term memory) 등의 딥러닝 기술들을 활용한 시도가 좋은 예이다[그림 4]³².



그러나 딥러닝 기술의 활용에 관해서는 신규 학자들 간에 미묘한 입장 차이가 있어 보였다. 젊은 연구자들은 딥러닝 등 새로운 기술의 활용에 상당히 적극적이었지만 원로 연구자들은 여전히 논리와 알고리즘에 많은 관심을 두고 있다는 느낌이었다. 이런 차이는 연구자의 소속기관이나 지역에 따라서 나타나기도 했는데, 예컨대 미국의 연구자들이 유럽의 연구자들보다, 기업에 소속된 연구자들³³이 대학에 소속된 연구자들보다 새로운 기술에 더 개방적인 것 같았다. 딥러닝 기술은 예측(prediction)에 강점을 보이지만, 이른바 블랙박스(black box)라는 표현에 나타나듯 그러한 결론에 이른 이유를 설명하지 못하는 문제가 있다. 법률 인공지능 분야는 단순히 예측만이 아니라 설명(explanation)까지도 요구하는 경우가 많다는 점을 고려하면 딥러닝 기술의 이용에는 어느 정도 한계가 있겠지만, 그럼에도 불구하고 최근 법률 서비스 분야의 생산성 향상에 가장 큰 역할을 하고 있는 것이 현실이다.

개인적으로는 예측을 위한 시스템과 설명을 위한 시스템을 별도로 구현하여 상호작용을 하도록 한다면 한 차원 높은 서비스

제공이 가능해질 수 있지 않을까 하는 생각이다. 인지과학의 성과에 따르면 인간 역시 직관적인 시스템 1과 분석적인 시스템 2를 보완적으로 활용하고 있다고 하지 않는가^{*34}. 이번 컨퍼런스에서도 이를 시사하는 발표가 있었다. 카네기 멜론 대학의 연구팀은 영업비밀 사건의 판결 예측에 관한 발표에서 “충분한 해당 분야의 지식과 잘 정의된 논증 도식이 주어진다면, 컴퓨터는 그 영역의 바탕에 있는 가치들을 고려하여 지적인 법률적 주장을 생성할 수 있을 뿐 아니라, 기계학습 알고리즘을 가지고 사건의 결과를 경쟁력 있게 예측하기 위하여 이러한 주장들을 활용할 수 있을 것”이라는 가설을 제시하였다. 위 연구팀은 발표 후 이어진 토론에서 “우리가 목표로 하는 것은 예측하고, 주장하고, 배울 수 있는 법률 인공지능(a legal AI that can predict, argue, and learn)”이라고 포부를 밝히기도 하였다^{*35}.

COLIEE

본 컨퍼런스와 함께 열린 행사 가운데 특히 필자의 주목을 끈 것은 올해로 네번째를 맞은 일종의 경연대회인 COLIEE(Competition on Legal Information Extraction and Entailment)^{*36}였다. 그 이유는 위 행사를 조직한 주요 인물 중 한 명이 한국계(앨버타 대학 김미영 교수)라는 점, 우리나라와 마찬가지로 대륙법계에 속하고 법조인 양성 과정이 유사한 일본의 변호사시험 문제를 푸는 것이 경연에서 주어진 과제라는 점, 그리고 무엇보다 우리나라에서 출전한 팀(인텔리콘 연구소, 대표 임영익 변호사)이 우승을 했다는 점 때문이다.

COLIEE는 일본 사법시험 문제에서 추출한 Yes/No 지문에 대하여 그 해결에 필요한 법조문을 제시하는 제1과제와 Yes/No 지문에 대하여 답까지 제시해야 하는 제2과제로 이루어져 있다. 제1과제가 정보 추출의 문제(legal information retrieval)라면 제2과제는 관련된 법조문과 지문을 비교하여 정답 여부까지 알아내야 하는 문제(recognizing textual entailment)이다. 트레이닝 데이터로는 2006년부터 2015년까지의 일본 변호사시험 민법 문제에 등장한 ‘지문 - 관련 법조문 - (제2과제의 경우) 정답 여부’ 세트가 주어지며, 테스트는 2016년의 변호사시험 문제에 의하여 이루어졌다. 출품된 시스템은 인간의 개입 없이 자동적으로 위 과제를 수행해야 하며 그 결과는 객관적 기준(recall, precision, f-measure, correctness)에 의하여 평가된다. 과제의 성격상 기계학습 기술이 전면적으로 동원될 수밖에 없는데, 트레이닝 데이터가 언어적·규범적인 형태이고 그마저도 매우 적다는 점 때문에 형태소 분석 등 선처리(pre-processing) 절차부터 효율적인 학습을 위한 전략 마련까지 만만치 않은 작업이 요구된다^{*37}. 우승 팀인 인텔리콘 연구소는 일본 민법의 준용 규정(mutatis

mutandis)을 효과적으로 활용하였다^{*38}. 개인적으로는 캐나다 팀의 접근방법이 인상적이었다. 캐나다 팀은 먼저 교착어(agglutinative language)인 한국어의 형태소 분석에 관한 기존의 연구 성과^{*39}를 활용하여 구문론적 구조(syntactic tree)를 의미론적 표현(semantic representation)으로 대치함으로써 문제를 ‘쉬운 문제’와 ‘어려운 문제’로 분류하였다. 그리고 쉬운 문제의 답(entailment)은 의미론적 표현에 의하여 찾고, 어려운 문제의 답은 일단 비지도학습(unsupervised learning)에 의하여 군집화한 뒤 쉬운 문제의 답(entailment)을 대응시키는 2단계 기법을 활용하여 좋은 성과를 내고 있었다^{*40}.인텔리콘 연구소가 수상을 하여 자랑스럽기도 했지만, 한편으로는 아쉬운 마음도 들었다. 대회에 참가한 여덟 팀 가운데 일본 팀이 넷이나 되었고 모두들 대학이나 연구소에 소속되어 정부의 지원을 받고 있었던 반면에 우리나라에서는 민간 업체 하나가 혼자서 힘겨운 싸움을 벌이고 있었기 때문이다.

소통과 협업, 그리고 정부의 역할

전통적으로 컴퓨터과학자들의 무대였던 법률 인공지능 분야는 이제 기업과 정부, 그리고 법률가와 소통하며 관심 영역을 확장해 가고 있다. 인공지능 기술이 사회와 경제에 미치는 영향이 점증하는 현실을 고려해본다면 이는 필연적인 동시에 바람직한 것이기도 하다. 인공지능과 법의 상호작용을 다루는 분야는 기술적·산업적 측면과 법적·정책적 측면을 모두 포용할 수 있어야 한다^{*41}. 다시 말해 기술-산업-법률-정책 등 4개의 영역에서 컴퓨터과학자 - 기업가 - 법학자 및 법률가 - 정부 및 정책전문가들이 서로 대화하고 소통함으로써만 문제를 해결할 수 있고 더 나은 세상을 만들어갈 수 있는 것이다. ‘AI and Law’라는 오래된 이름은 이제야 그에 걸맞은 내용을 가지게 된 것인지도 모른다.

안타깝게도 우리나라의 현실은 아직 이러한 이상적인 모습에 미치지 못하고 있다. 인공지능 분야를 전공한 컴퓨터과학자를 비롯한 전문가가 태부족인 실정이며, 특히 법률 인공지능 분야의 전문가는 손으로 꼽을만한 수준이다. 상황이 이렇게 된데는 무엇보다 지난 몇 년간 기초적인 과학기술과 학문에 대한 지원을 외면한 정부의 탓이 크다. 많은 법률가들 역시 인공지능 기술에 의하여 자신들의 입지가 줄어들까봐 막연히 두려워하기만 할 뿐 새로운 기술이 안겨다 줄 기회는 알아채지 못하고 있다. 필자가 보기에는 법률 서비스 분야야말로 인공지능 기술에 의하여 가장 많은 생산성 향상이 이루어지고 가장 큰 시장 확대가 이루어질 수 있는 분야이다. 이제부터라도 과학자와 기업가, 법률가와 정책 전문가들이 만나 서로 대화를 시작해야 할 것이다. 필자가 참여하고 있는 한국인공지능법학회가 그러한 무대 중 하나가 되기를 기대해 본다.

^{*1} 참고 | 더글라스 호프스태터(박여성 역), “괴델, 에셔, 바흐”(상·하), 까치, 1999. ^{*2} 참고 | Hans Moravec, “Mind Children : The Future of Robot and Human Intelligence”, Harvard University Press, 1990. ^{*3} 참고 | 레이 커즈와일(장시형·김영남 역), “특이점이 온다 - 기술이 인간을 초월하는 순간”, 김영사, 2007. ^{*4} 참고 | Nick Bostrom, “Superintelligence: Paths, dangers, strategies,” Oxford University Press, 2014. 위 책은 올해 4월 도서출판 까치에서 “슈퍼인텔리전스 경로, 위험, 전략”이라는 제목으로 번역본(조성진 역)이 출간되었다. ^{*5} 참고 | www.mendeley.com “A.I. and Law” ^{*6} 참고 | 통상 ‘Law Courts’라고 불리며 고등법원 격인 ‘High Court’와 민사 항소법원인 ‘Court of Appeal of England and Wales’가 있는 건물이다. ^{*7} 참고 | https://techindex.law.stanford.edu/statistics ^{*8} 참고 | https://nms.kcl.ac.uk/icail2017/ ^{*9} 참고 | Katie Atkinson. AI and Law in 2017: Turning the hype into real world solutions. (IAAIL Presidential Address). 이하 편의 상 ICAIL 2017의 Proceeding은 발표자와 제목만을 적기로 한다. ^{*10} 참고 | Katie Greenwood, Trevor Bench Capon, and Peter McBurney. “Towards a computational account of persuasion in law.” Proceedings of the 9th international conference on Artificial intelligence and law. ACM, 2003. 그 기본 도식은 다음과 같다. ① (A5) In the Current Circumstances R ② we should perform Action A ③ to achieve New Circumstances S ④ which will realize some goal G ⑤ which will promote some value V. 여기에서 가치는 왜 G가 목적 인지를 설명해주며 A라는 행동을 하는 이유가 된다. ^{*11} 참고 | Dan Cartwright and Katie Atkinson. “Using computational argumentation to support e-participation.” IEEE Intelligent Systems 24.5 (2009). ^{*12} 참고 | Adam Wyner, Trevor Bench-Capon, and Katie Atkinson. “Arguments, values and baseballs: Representation of Popov v. Hayashi.” JURIX. Vol. 165. 2007. ^{*13} 참고 | https://sites.google.com/a/policy-impact.eu/public/home. . ^{*14} 참고 | Al-Abdulkarim, Latifa, Katie Atkinson, and Trevor JM Bench-Capon. “Abstract Dialectical Frameworks for Legal Reasoning.” JURIX. 2014. ^{*15} 참고 | Al-Abdulkarim, Latifa, et al. Angelic Environment: Support for the Construction of Legal KBS. Technical Report ULCS-17-002, University of Liverpool, 2017; http://cgisci.liv.ac.uk/~tbc/publications/LAKASATBCfinal.pdf ^{*16} 참고 | Rohan Nanda, Luigi Di Caro, Guido Boella, Hristo Konstantinov, Tenyo Tyankov, Daniel Traykov, Hristo Hristov, Francesco Costamagna, Llio Humphreys, Livio Robaldo, Michele Romano. A Unifying Similarity Measure for Automated Identification of National Implementations of European Union Directives. ^{*17} 참고 | SMARTLAW: Workshop on Blockchain, Smart Contracts and Law. ^{*18} 참고 | ICAIL 2017 Workshop on Evidence & Decision Making in the Law. ^{*19} 참고 | Argyro Karanasiou, Dimitris Pinotis, “Towards a Legal Definition of Machine Intelligence: The Argument for Artificial Personhood in the Age of Deep Learning” ^{*20} 참고 | Henry Prakken, “On Making Autonomous Vehicles Respect Traffic Law: a Case Study for Dutch Law” ^{*21} 참고 | Maja Brkan. AI-Supported Decision-Making under the General Data Protection Regulation. ^{*22} 참고 | Karen Yeung, “Using AI systems to personalise, predict and automate the application of law: A fundamental assault on the concept and rule of law?” ^{*23} 참고 | Chen, Daniel L., Adithya Parthasarathy, and Shivam Verma. “The Genealogy of Ideology: Predicting Agreement and Persuasive Memes in the US Courts of Appeals.” (2016). ^{*24} 참고 | Henry Prakken, “On Making Autonomous Vehicles Respect Traffic Law: a Case Study for Dutch Law” ^{*25} 참고 | https://law.stanford.edu/codex-the-stanford-center-for-legal-informatics/ ^{*26} 참고 | Prakken, Henry, and Marek Sergot. “Contrary-to-duty obligations.” Studia Logica 57.1 (1996): 91-115. ^{*27} 참고 | Xavier Parent, Leendert van der Torre. The pragmatic oddity in norm-based deontic logics. ^{*28} 참고 | Vern R. Walker, Ji Hae Han, Xiang Ni and Kaneyasu Yoseda. Semantic Types for Computational Legal Reasoning: Propositional Connectives and Sentence Roles in the Veterans’ Claims Dataset. ^{*29} 참고 | Matteo Cristani, Francesco Olivieri, Antonino Rotolo. Changes to Temporary Norms. ^{*30} 참고 | Celia da Costa Pereira, Beishui Liao, Alessandra Malerba, Antonino Rotolo, Leendert van der Torre. Combining Fuzzy Logic and Formal Argumentation for Legal Interpretation. ^{*31} 참고 | Juliano S.A. Maranhao. A logical architecture for dynamic legal interpretation. ^{*32} 참고 | James O’ Neill, Paul Buitelaar, Cecile Robin, Leona O’ Brien. Classifying Sentential Modality in Legal Language: A Use Case in Financial Regulations, Acts and Directives. ^{*33} 참고 | Jack G. Conrad, Khalid Al-Kofahi. Scenario Analytics. Analyzing Jury Verdicts to Evaluate Legal Case Outcomes. 유명한 법률정보 검색 서비스인 LexisNexis를 운영하는 Thomson Reuters 에서 주도한 연구이다. 위 회사는 지난 2015. 10. IBM과 Watson 기술을 법률정보검색에 활용하는 데 활용하기로 하는 파트너십을 구축하여 화제가 되었는데(https://www.legaltechnology.com/latest-news/thomson-reuters-announces-ibm-watson-partnership/), 우리나라 최대의 민간 법률정보회사인 로앤비를 인수하여 운영하고 있기도 하다. ^{*34} 참고 | 대니얼 카너먼(이진원 역), “생각에 관한 생각”, 김영사, 2012. ^{*35} 참고 | Matthias Grabmair. Predicting Trade Secret Case Outcomes using Argument Schemes and Learned Quantitative Value Effect Tradeoffs. ^{*36} 참고 | https://www.nms.kcl.ac.uk/icail2017/cfoliee.php; http://webdocs.cs.ualberta.ca/~miyoung2/COLIEE2017/ ^{*37} 참고 | 법률 분야의 특성상 주석이 달린(annotated) 양질의 트레이닝 데이터 부족은 만연한 현상이다. 이번 컨퍼런스에서는 이를 해결하기 위한 방안도 발표되었는데, 주석이 달린 소규모의 리걸 온톨로지(legal ontology)와 주석이 없는 대규모의 리걸 온톨로지를 위키피디아를 이용하여 연계시킴으로써 저렴한데 데이터를 구축하는 방안이 그것이다. 비교법적 연구에 상당한 도움을 줄 수 있는 잠재력이 있다고 생각되는 방안이다. Cristian Cardellino, Milagro Teruel, Laura Alonso Alemany, Serena Villata. A Low-cost, High-coverage Legal Named Entity Recognizer, Classifier and Linker. ^{*38} 참고 | Seongwan Heo, Kihyun Hong, Young-Yik Rhim. Legal Content Fusion for Legal Information Retrieval. ^{*39} 참고 | Kim, Mi-Young, Sin-Jae Kang, and Jong-Hyeok Lee. “Resolving Ambiguity in Inter-chunk Dependency Parsing.” NLPRS. 2001. ^{*40} 참고 | Mi-Young Kim, Randy Goebel. Two-step Cascaded Textual Entailment for Legal Bar Exam Question Answering. ^{*41} 참고 | 이상용, “인공지능과 법 - 범위와 방법,” 대전지방법원-충남대학교 법학전문대학원-한국인공지능법학회 공동 주관 공동세미나(인공지능의 시대, 위 기 그리고 기회), 2017.

AI 접목된 의료영상의 주요 플레이어들

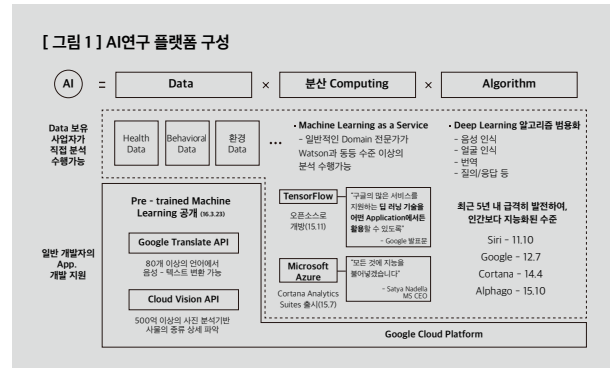
의료산업에서 AI 적용에 대한 관심은 뜨겁다. 가장 데이터가 많은 곳이기도 하고 직접적으로 효과가 나타날 곳이기도 하기 때문이다. 그러다보니 병원과 의료기기 회사, 제약 회사와 같은 전통적인 사업자들도 바빠 움직이지만 스타트업들도 많이 나타나 주목받고 있다. 주요 스타트업들을 조사하여 발표하는 cbinsights.com에 나오는 의료 AI 기업들이 작년 3분기에 70여 업체가 조사되었는데 올해는 벌써 100여 개를 넘어섰다.

인공지능 기반의 의료영상 시장 현황:

주요 플레이어와 관련 기술력 소개를 중심으로

시장 전망에서 나오는 숫자들이란 것이 실감하기 쉽지 않지만 한 기관의 자료만 인용해보자. 프로스트앤셀리반에서는 2014년에 약 6.3억 달러의 의료 AI 시장이 2021년까지 10배 이상 성장하여 66억 달러가 넘을 것이라고 전망했다¹. 실제 몇 천만 달러 수준이던 의료 AI 투자액은 2014년부터 몇 억 달러 단위로 넘어갔다. 작년 1분기 AI 투자액 중에서는 15%가 의료 분야로 투자되기도 했다². 이는 딥러닝의 영향이라.

의료 AI 영역에서 이전부터 회사가 많았던 환자 데이터 관리 분야나 리스크 분석 분야에 비해 가장 빨리 증가하고 있으며 현재 대세를 차지하고 있는 스타트업들은 의료 영상을 이용한 진단 지원분야다. 이렇게 스타트업들이 많이 나타나는 데는 딥러닝이 현실 적용에 유리한 경험적 연구(empirical research)의 특성을 지니기 때문이다. 알고리즘이라는 것, 네트워크를 설계한다는 것은 기본적인 이론을 바탕으로 잘 알려진 여러 네트워크 요소들을 데이터의 특성에 맞추어 마치 레고블럭 쌓듯 쌓아가는 과정이다. 이렇게 만들어진 네트워크를 돌리기 위한 컴퓨팅 파워, 개발 플랫폼은 클라우드로 제공되고 있다. 실제 데이터만 잘 구할 수 있으면, (실은 그게 제일 어려운 부분이기도 하지만) 누구라도 한번 도전해볼 만한 시장이다. 더군다나 병원이라는 곳은 매일 매일 엄청난 양의 (영상) 데이터가 생겨나는 곳이 아닌가? 새로운 데이터만 구할 수 있으면 아래 그림처럼 이미 “as a service”로 분석할 수 있는 환경이 되었다.



더군다나 영상 인식이라고 하는 것은 딥러닝 분야 중에서도 가장 많이 발전됐고 새로운 논문들이 쏟아져 나오는 영역인데다 CNN(Convolutional Neural Network) 기법에 기초한 알렉스넷이니 유넷이니 인셉션이니 하는 증명된 네트워크도 손쉽게 적용해 볼 수 있는 분야이니 말이다.

그래서 장기적으로는 새로운 모델을 만들어서 알고리즘의 혁신을 꿈꾸면서 단기적으로는 기존에 나와있는 CNN 모델을 데이터에 잘 적용하여 좋은 결과를 찾아내는 점진적인 방법이 병행되고 있다. 우선 몇몇 스타트업들을 살펴보자.

Enlitic

2014년에 설립된 이 회사는 MIT Technology Review에서 대표적인 의료 AI 기업으로 소개되었다. 주 연구 분야는 엑스레이와 같은 영상 데이터에 딥러닝을 적용하여 병의 유무와 병변의 위치를 찾는 것이다. 이 기업에 대해 주로 소개되는 내용은 흉부 엑스레이에서 폐암과 관련된 결절의 위치를 찾아내는 것과 정형외과 엑스레이에서 골절 부위를 찾아내는 것이다. 2016년 이코노미스트에서 악성 종양을 의사보다 50% 더 잘 찾아낸다고 보도해 화제가 되었다³. 호주 의료 영상 서비스 기업과 협력하여 40여개 병원에서 테스트를 하고 있으며 약 100억원 규모의 투자를 받았다. 최근에는 CNN 기반의 영상 외에도 자연어 처리(NLP) 기능을 영상 진단 플랫폼에 추가할 계획이라고 한다.

Butterfly Network

이번에는 좀더 멀리 보고 있는 기업이다. 2011년에 로쓰버그(Jonathan Rothberg) 박사와 MIT 연구자들이 모여서 스타트렉에 나오는 트라이코더처럼 들고 다니면서 인체 내부를 컬러 영상으로 살펴볼 수 있는 스마트폰 크기의 영상 기기를 만들겠다고 시작한 기업이다⁴. 천억 원대의 투자를 받았고 몇 년 전에 1-2년 안에 상품을 내겠다고 했지만 아직 개발중이다. 로쓰버그 박사는 이미 초음파 기술이 심장 속까지 들여다보고 있는데 아직도 1800년대에 나온 청진기에 진찰을 의존하는 현실을 안타까워했다. 특히 의료영상 서비스를 받지 못하는, 다수의 개발도상국에 희망을 주고자 한다고 말했다. 초음파 영상으로 학습된 딥러닝 네트워크를 하드웨어칩에 내장하고 데이터는 클라우드에 올려서, 환자 영상을 찍으면 찍을수록 더욱 진화하는 기기를 구상하고 있다.

Arterys

스탠포드 대학교 출신들이 모여 출발한 이 회사는 지이헬스케어(GE Healthcare)와 협력하여 심장 영상을 다양한 형태로 보여줄 수 있는 기술을 개발하고 있다. GE와 협력하여 기존에 60분 가량 걸리던 심장 MRI를 10분 이내로 줄이고 3차원 영상에 더해서 혈류의 방향과 속도까지 표시해준다. 구체적으로 밝히지는 않지만 2차원 영상에서 3D 영상을 정합하여 만들어내고 심방과 심실을 자동으로 나누고, 혈류량을 측정하는 등 유동과 기능을 분석하기 위해서 딥러닝을 사용했다고 한다. 딥러닝과 직접적이지는 않지만 의료 영상 데이터를 영상촬영장비에서 바로 클라우드로 보내고 인터넷을 통해 시각화한 결과를 보여주는 점도 주목할 만하다. 올해 초에 FDA 허가를 받았으며 GE MRI에 탑재되어 미국 Fairfax Radiology Consultants에서 임상 적용해 쓰고 있다.

글 | 하영식 ysha@vaiim.com

ICT 및 healthcare 영역에서 전략 컨설팅을 하는 바임컨설팅(www.vaiim.com)의 대표파트너이다. 인간, 컴퓨터, 네트워크라는 세 키워드를 마음에 품고 사회에 나왔는데 어느덧 이게 현실이 되는 세상이 온 것 같아, 설레는 마음으로 인공지능이 바꾸어갈 세상에 대해 공부하고 세상을 한 발짝이라도 좋게 바꾸어 보려고 노력하고 있다.

우리 나라에도 다른 나라와 비교하여 손색없는 수준으로 비슷한 기술을 개발하고 있는 기업들이 있다. 대표적인 회사가 뷰노(Vuno)와 루닛(Lunit)이다.

Vuno

삼성종합기술원과 포항공대 출신들이 주축이 되어 만든 회사로 엑스레이, CT, MRI 영상에서 진단을 자동화하는 연구를 하고 있다. 흉부 엑스레이에서 결절을 찾는 연구 결과를 발표하고 손목 엑스레이에서 골연령을 추정하는 연구 결과도 발표했다.

최근에는 딥러닝이 경험적 연구에 기반한 블랙박스라는 특성으로 인해 의사들이 수용하기 어려운 현실을 고려하여 어텐션맵 (attention map) 등을 이용, 영상의 어느 지점이 의사결정에 중요한 영향을 끼쳤는 지를 보여주어 딥러닝 결과를 해석할 수 있는 방안에 대한 연구를 진행하고 있다. 또 영상뿐 아니라 음성, 텍스트 기록 등과 연계한 멀티모달 (multi-modal) 연구도 병행하고 있다.

Lunit

이미지넷 대회 등으로 딥러닝 분야에서 우리나라에서보다 세계적으로 먼저 알려진 기업이다. 마찬가지로 흉부 엑스레이 연구와 함께 최근에는 유방촬영술(mammography) 영상을 이용한 유방암 진단, 병리 영상에 대한 진단을 주요 분야로 연구를 진행하고 있다. ILSVRC, MICCAI와 같은 영상기술 관련 알고리즘 대회에서 좋은 성과를 보여 주목을 받고 있으며 58억원 규모의 투자를 받았다. 특히 병리 영상은 이미지 한 장이 몇십 GB에 달할 정도로 커서 일반 영상에서 쓰였던 알고리즘을 적용하기 위한 전처리 과정이 매우 복잡하고 연구가 필요한 분야이기도 하다. 또한 딥러닝을 이용한 진단 알고리즘뿐만 아니라 병원에서 사용하기 쉽도록 시각화, 편집 등 UI 개발 및 PACS 연동에도 신경을 많이 쓰고 있다.

그러면 초대형 기업들(software giants)은 뭘하고 있을까?

Google

위에 열거한 스타트업들과 비슷한 일을 구글 스케일로 하고 있다. 작년 말에 대표적인 의료 학회인 JAMA (the journal of the American medical association)에 논문이 실려 관련업계에서는 큰 뉴스가 되었던 당뇨병 망막병증 연구를 살펴보자⁵. 인도에서 안과 전문의 54명이 6개월 정도 참여하여 12만여 장의 망막안저영상을 분석했다. 결과를 테스트하기 위해서도 전문의 자격증을 가진 안과 의사 여러 명이 검토했다. 올 초에는 유방암 환자의 병리 조직을 진단하는 프로젝트를 수행했다. 인셉션 네트워크를 활용했는데 시간 제한 없이

검토했던 병리학자에 비해 더 좋은 결과가 나왔다고 한다⁶. 구글이 단순히 연구만 하고 있을 것 같지는 않다. 안저영상 연구를 인도에서 한 게 특히 의미가 있는데 선진국에서는 이미 병원에서 쉽게 안저영상 검사를 할 수 있지만 개발도상국들은 그렇지 않다. 구글이 자동화된 안저영상 검사/판독 시스템을 인도에 보급한다면 인간 의사가 새로 들어갈 틈은 별로 없을 것이다.

IBM

너무 유명한 Watson이다. AI로 유명세를 떨쳤지만 딥러닝과는 조금 다른, 대화 기반으로 문헌을 분석하는 시스템으로 유사 사례를 분석하는데 강점이 있어 의사의 진단을 보조하는 시스템으로 효과가 있다. 우리 나라 몇몇 병원에서 도입된 종양진단 프로그램 (Watson for Oncology)도 미국 MSK (Memorial Sloan Kettering) 병원의 진단 프로세스에 기반하여 진료법을 추천하는 것이다. Watson도 2017년 올란도에서 열린 대표적인 의료 정보학회인 HIMSS에서 영상 분석 솔루션을 내놓았다. 앞에서 보인 기업들이 영상 자체를 판독하는데 집중한다면 Watson은 텍스트 등의 EMR 데이터를 해석하여 진단 결과에서 차이점을 찾고 잘못된 결과가 있으면 이를 수정할 수 있도록 하는, 원래 자신이 잘 하던 분야에서 영역을 확장해 가고 있다. 또 하나 주목할 것은 의료 영상 데이터를 얻기 위한 협력구도이다. 전세계적으로 24개의 기관이 참여하는 의료영상협의체를 운영하는데⁷ 의료 영상 데이터 확보에 목적을 두고 2015년에 1조원이 넘는 돈을 주고 인수한 Merge Healthcare도 포함된다.

엑스레이, CT, MRI 같은 의료영상과는 조금 다른 영상 활용 시도도 살펴보자.

AiCure

위 사례들보다는 상대적으로 단순하고 일반화된 인식기술을 활용한다. 영상인식을 환자의 얼굴인식과 약을 식별하는데 사용한다. 환자가 스마트폰 앞에서 약을 보여주고 인식이 되고 나면 입에 넣은 모습과 함께 그 다음에 삼킨 것까지 영상촬영을 해서 약을 제대로 먹었는지 파악한다. 단순히 약을 먹는 것을 인식하는 것이 아니라 의사-환자-보험기관이 함께 쓰는 플랫폼을 제공해서 처방된 약을 잘 먹고 있는지 의료 기관에서 실시간으로 관리하고 환자에게는 복약 시간 안내도 같이 한다⁸.

대표적인 사용처는 임상시험에서 시험참가자들에 대한 모니터링이고, 또 만성질환자의 경우 약만 제대로 먹어도 예후가 크게 달라진다고 하니 이들에 대한 지속적인 관리는 요즘같이 의료비용이 계속 늘고 있는 상황에서 크게 관심을 받는 분야다.

Stanford AI-assisted care

이미지넷을 만든 페이페이리 (Feifei Li) 교수가 이끌고 있는 컴퓨터비전과 스탠포드 의과대학의 공동연구다. 병원감염의 해결책으로 손씻기가 중요하지만 의료 현장에서 잘 이루어지지, 어떻게 하면 더 나아지게 할 수 있는지를 연구하기 위해 병동 전역에 카메라를 설치했다. 개인정보 보호를 위해 외형을 식별할 수는 없는 뎁스카메라를 사용하는데 주로 어떤 활동 다음에 씻지 않는지, 동선상에 문제가 없는지 등 병원의 표준 매뉴얼과 실제 활동 사이의 차이를 보고자 한다.

또한 독거노인의 움직임, 식사, 걸음걸이, 휴식 등 일상활동 패턴을 분석하고 위급한 일이나 일상에 대한 요약을 환자 가족에 알려주는 연구, 중환자실(ICU)에서 영상과 신호 데이터를 이용해 환자 건강 상태를 확인하고 간호사의 활동을 해석하여 자동으로 간호 업무 내용을 텍스트로 기록하는 등에도 딥러닝 기반의 영상 분석을 활용한다⁹.

정리하며

의료영상 분야는 첫째, 이미 많이 쓰이고 있는 영상분석 알고리즘 또는 네트워크를 잘 활용할 수 있다는 점에서 접근이 쉽다. 둘째, 다른 EMR 데이터에 비해 상대적으로 잘 표준화된 데이터를 구하기 쉽다는 점에서 많은 기업들이 비슷한 기술로 뛰어들고 있다. 최근 추세로는 몇 개 사례에서 보았듯이 영상과 다른 EMR 데이터의 결합, 의사의 음성 데이터와의 결합, 유전체 분석 데이터와의 결합 등 멀티모달 분석에 대한 시도가 있다. Relational Network 같은 기술적 발전들이 여러 데이터의 통합에 도움을 주지 않을까 기대해 본다.

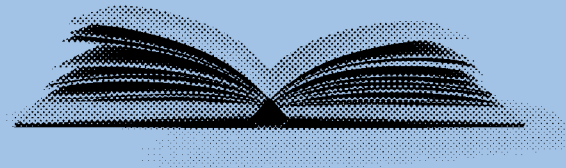
영상진단이 아니라 진단 전체로 넓게 보면 의료 서비스가 쉽지 않은 곳에서 환자의 자가진단을 도와주는 챗봇 형태의 서비스인 Babylon이나 Your.MD, Ada Health 같은 기업도 주목할 만하고 영상, 검사결과, 라이프로그, 유전정보 등을 모두 모으겠다는 iCarbonX 같은 중국 회사가 6천억원대의 투자를 받고 바빠 움직이고 있다¹⁰.

하지만 영상진단이 단순히 영상이 많다고 진단이 되는 것이 아니라 대부분의 경우 잘 레이블된 (정답이 있는) 고품질 데이터를 의료 전문가인 의사들이 상당한 시간을 들여서 만들어야 한다는 점에서 막상 계속하기는 쉽지 않은 분야다. 그리고 각국의 보험제도와 연관되어 있는 의료산업의 특성상 의료기기에 연계해서 판매하거나 보험수가를 받지 못하면 제대로 된 수익모델을 만드는 것도 쉬운 일은 아니다. 잠재적인 환자 입장에서 어쨌든 AI 기술이 더 발달하는 것은 기대되는 일이지만 말이다.

¹ 참고 | Frost & Sullivan, "Transforming Healthcare through Artificial Intelligence System, 2016" ² 참고 | <https://www.cbinsights.com/blog/artificial-intelligence-healthcare-funding/> ³ 참고 | <https://www.economist.com/news/special-report/21700758-will-smarter-machines-cause-mass-unemployment-automation-and-anxiety> ⁴ 참고 | <https://www.technologyreview.com/s/532166/with-100-million-entrepreneur-sees-path-to-disrupt-medical-imaging/> ⁵ 참고 | <http://jamanetwork.com/journals/jama/article-abstract/2588763> ⁶ 참고 | <https://research.googleblog.com/2017/03/assisting-pathologists-in-detecting.html> ⁷ 참고 | <http://www.merge.com/News/Article.aspx?ItemID=677> ⁸ 참고 | <https://aicure.com/> ⁹ 참고 | <http://aicare.stanford.edu> ¹⁰ 참고 | <http://www.nature.com/news/chinese-ai-company-plans-to-mine-health-data-faster-than-rivals-1.21258>

AI 연구 동향과

강화학습 개념



learning

정수현 | AI 3대 학회의 논문 경향

32

엄태웅 | 딥러닝 연구의 현재와 미래 part 2

36

최성준, 이경재 | 알파고를 탄생시킨 강화학습의 비밀 part 2

42

NIPS와 ICML에서 2005년부터 2016년까지 발표된 논문 6,163건을 분석한 카카오 시리포트 5월호(Vol.3) 기억하시지요? 이 분석 결과는 발행 즉시 카카오 정책지원파트 브런치를 중심으로 소셜네트워크에 빠르게 확산됐습니다. 아직도 주요 AI 연구 모임에서는 이 결과가 회자됩니다. 이 분석 작업을 진행했던 정수현 님이 이번에는 올해 열린 CVPR, ICLR, ICML에서 발표된 연구물을 논문의 주요 단어 분석을 통해 정리했습니다. 최근의 AI 연구 흐름 확인을 위한 참고 자료로 챙겨보실 것을 권해 드립니다. 역시 카카오 시리포트 5월호에 함께 게재됐던 엄태웅 님이 정리한 주요 딥러닝 연구 논문 part2는 딥러닝 연구의 흐름을 확인하고자 하시는 분들에게 긴요한 정보를 드릴 겁니다. 알파고 두뇌의 주요 원리인 강화학습에 대한 최성준 님과 이경재 님의 강화학습 설명 글은 따로 정리해 두고두고 보실 자료로 권해드립니다.

AI 3대 학회의 논문 경향

과학 기술 발전의 진일보는 다양한 곳에서 이루어지고 있지만, 역사적으로 분야의 전문가들이 모이는 학회(Conference)를 통해 다양한 주제가 심도있게 논의되고 이후 발전에 큰 획을 긋는 경우들도 많이 있다. 20세기 물리학 발전에 가장 큰 계기가 된 학회는 1927년 Solvay 물리 학회를 뽑을 수 있다. 당시 양자역학 이론을 두고 아인슈타인, 닐보어, 마리큐리, 슈레딩거, 하이델베르크, 파울리 등 당대 최고의 물리학자들이 참석하여 열띤 토론을 벌였다. 이 역사적인 순간을 함께한 물리학자들 중 17명이 후에 노벨상을 수상하게 된다. AI연구에서도 손꼽히는 학회들이 앞으로 많이 등장 할 것이다. 그 중 하나가 될지도 모르는 2014년 캐나다 몬트리올에서 열린 NIPS(Neural Information Processing Systems)는 지난 카카오 AI 리포트 1호를 통해서 소개한바 있다(<https://brunch.co.kr/@kakaο-it/51>). 이번에는 2017년도 진행된 AI 및 머신러닝 학회에서 발표된 연구물의 동향을 논문 제목의 키워드를 중심으로 요약했다. 이번 글이 현재 어떤 연구들이 이루어질 수 있는지 돌아볼 수 있는 실효적인 참고자료가 되길 희망한다.

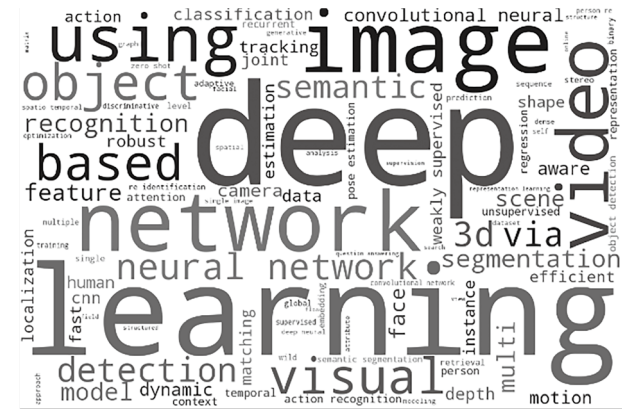
CVPR

[CVPR에 가장 많이 인용된 논문 5편]

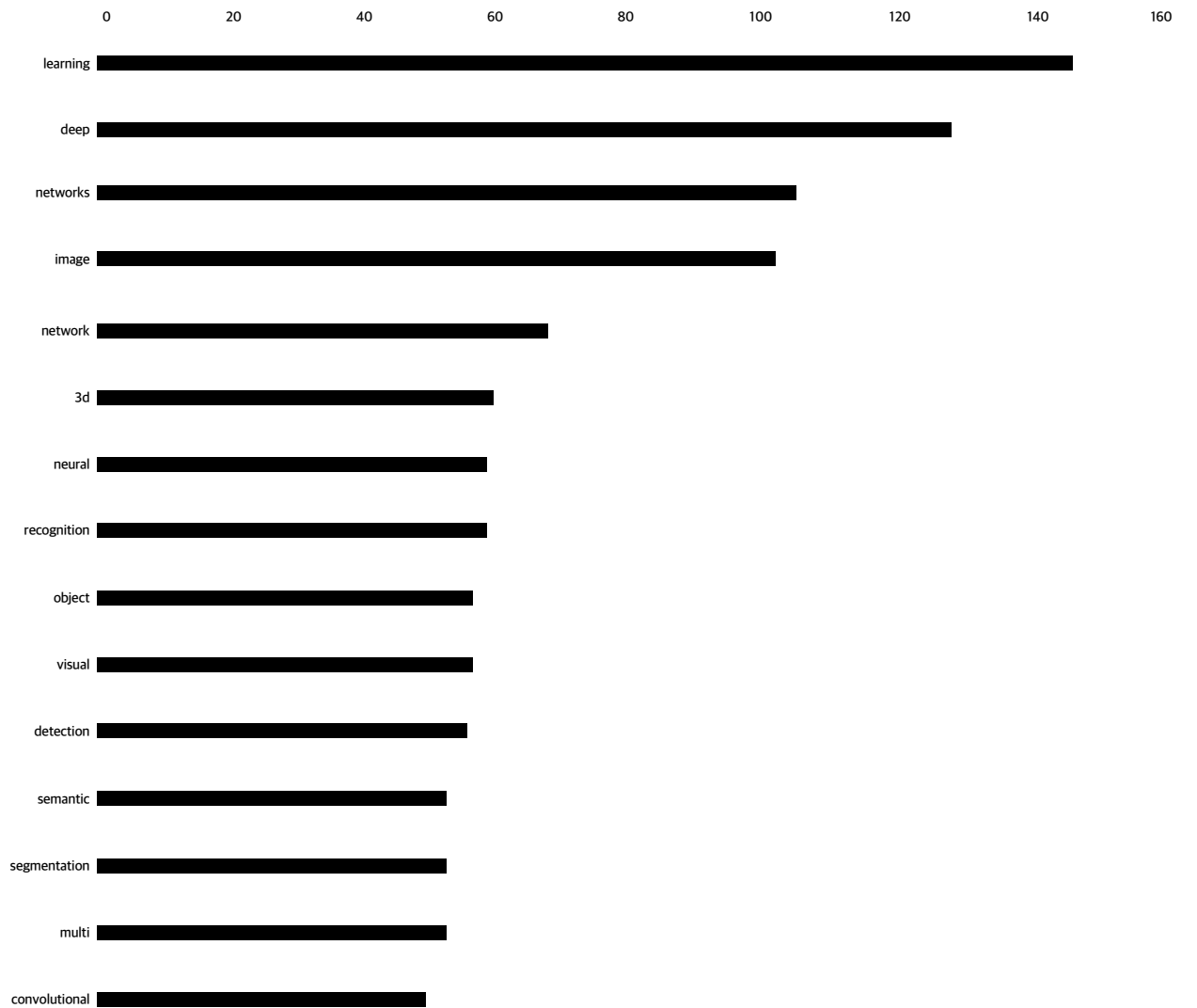
순위	논문	인용	저자
1	Densely Connected Convolutional Networks	155회	G Huang, Z Liu 외 2인
2	Image-To-Image Translation With Conditional Adversarial Networks	144회	P Isola, JY Zhu,외 2인
3	Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network	121회	C Ledig, L Theis 외 9인
4	YOLO9000: Better, Faster, Stronger	48회	J Redmon, A Farhadi
5	Aggregated Residual Transformations for Deep Neural Networks	44회	S Xie, R Girshick외 3인

2017년 8월 23일 기준

[CVPR 발표된 논문들 제목에 많이 등장한 핵심어]



[CVPR 발표된 논문들 제목에 가장 많이 등장한 단어들]



글 | 정수현 noah.jung@kakaοcorp.com

숨은의미? 머니볼의 힘? 숫자들에서 남들이 찾지 못한 숨겨진 의미를 찾아내는 걸 좋아한다. 어릴적 모바일 게임을 개발하던 코딩 실력으로 디퍼닝을 쫓아가려고 허덕거리며 달리고 있다. 하지만, 오래 달리는건 자신있다. 마라톤을 달리며 나와의 싸움을 이기는데 익숙해 있기 때문이다. 비록 늦게 출발했지만 언젠가는 트랙에서 함께 뛰고 있기를 기대하며, 지금도 비록 느리지만 달리고 있다.

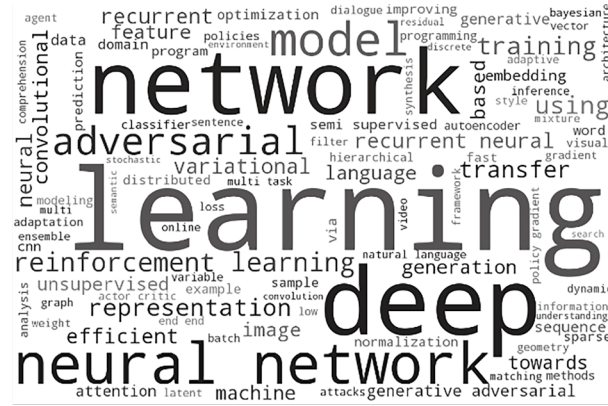
ICLR

[ICLR에 가장 많이 인용된 논문 5편]

순위	논문	인용	저자
1	Energy-based Generative Adversarial Networks	99회	J Zhao, Y LeCun 외 1인
2	Dynamic Coattention Networks For Question Answering	86회	C Xiong, V Zhong 외 1인
3	Machine Comprehension Using Match-LSTM and Answer Pointer	83회	S Wang, J Jiang
4	Adversarial Feature Learning	83회	J Donahue, P Krahenbuhl 외 1인
5	Bidirectional Attention Flow for Machine Comprehension	78회	M Seo, A Kembhavi 외 2인

2017년 8월 23일 기준

[ICLR 발표된 논문들 제목에 많이 등장한 핵심어]



ICML

[ICML에 가장 많이 인용된 논문 5편]

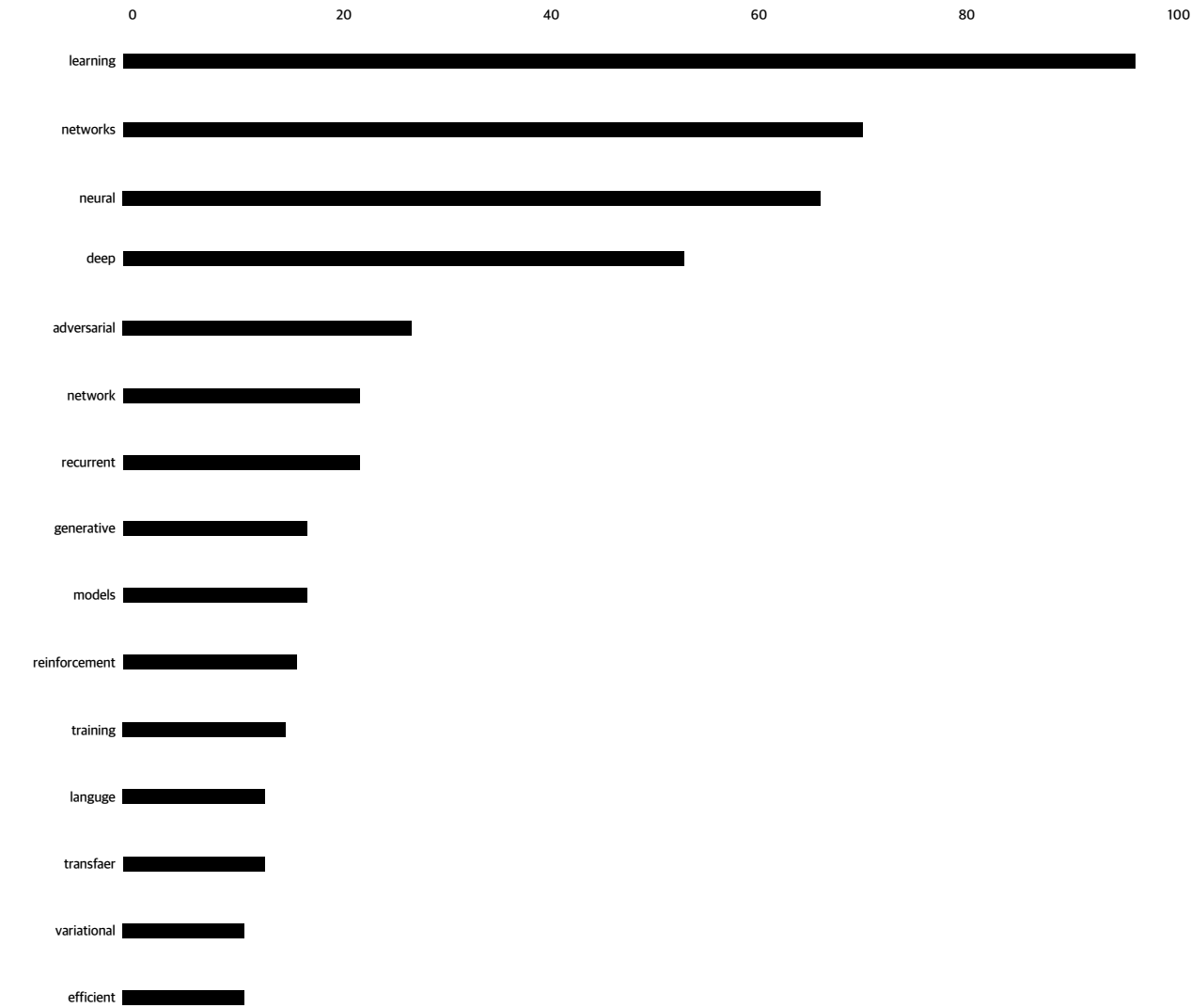
순위	논문	인용	저자
1	Conditional Image Synthesis with Auxiliary Classifier GANs	41회	A Odena, C Olah 외 1인
2	Recurrent Highway Networks	39회	JG Zilly, RK Srivastava 외 2인
3	Video Pixel Networks	25회	N Kalchbrenner, A Oord 외 5인
4	Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks	25회	L Mescheder, S Nowozin 외 1인
5	Programming with a Differentiable Forth Interpreter	20회	S Riedel, M Bosnjak 외 1인

2017년 8월 23일 기준

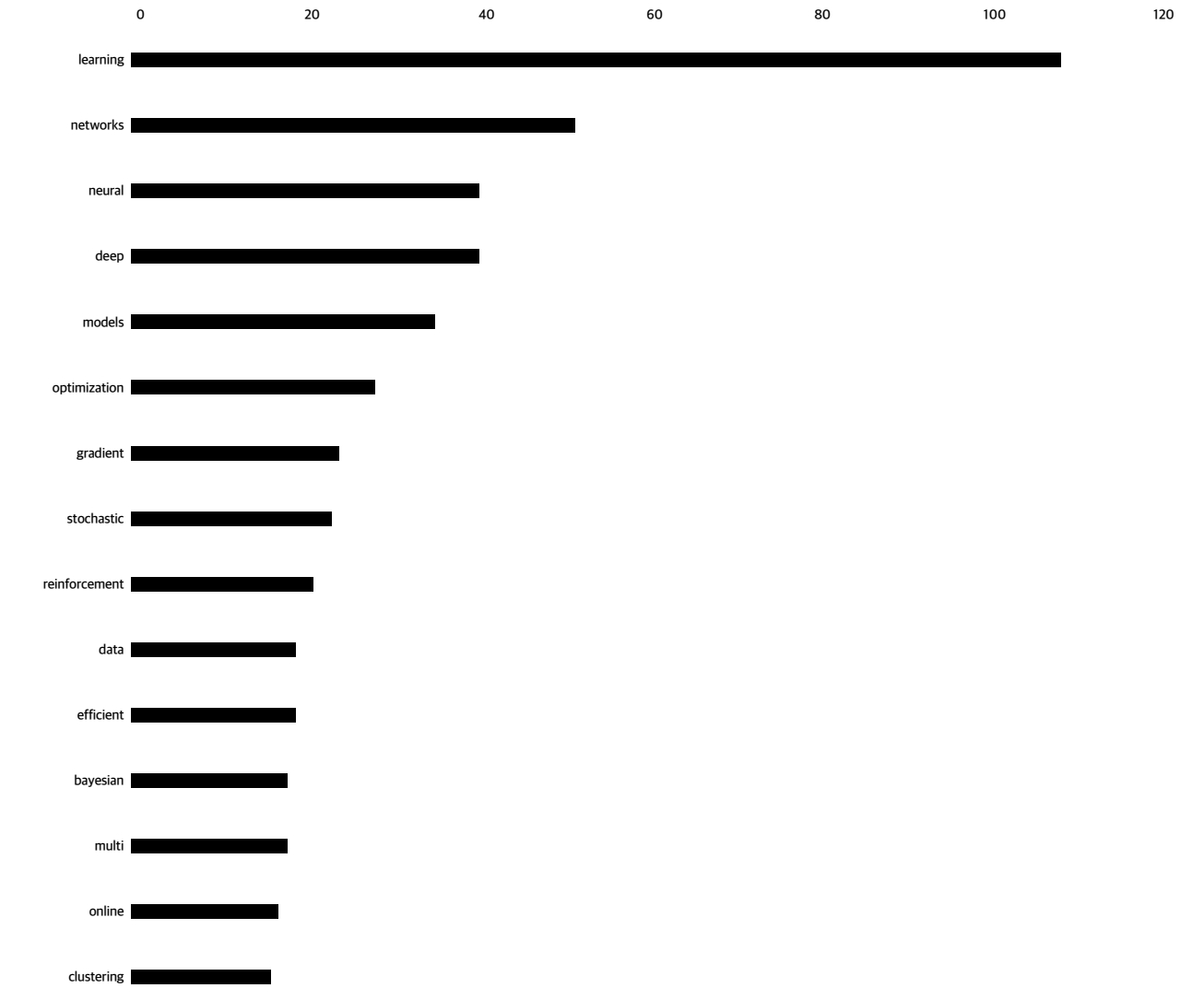
[ICML 발표된 논문들 제목에 많이 등장한 핵심어]



[ICLR 발표된 논문들 제목에 가장 많이 등장한 단어들]



[ICML 발표된 논문들 제목에 가장 많이 등장한 단어들]



딥러닝 연구의 현재와 미래 part 2. (2/2)

이제까지 딥러닝의 성공을 이끈 세가지 요소를 꼽는다면, 첫째, 사용가능해진 많은 양의 데이터, 둘째, 그것을 확장성(scalable) 있게 사용할 수 있게 한 깊은 신경망(deep neural network)의 개발, 마지막으로, 빅데이터 연산을 개인용 PC에서도 할 수 있게 만든 고성능 GPU의 발전(general-purposed graphics processing unit, GPGPU)을 꼽을 수 있다¹. 특히 “많은 양의 데이터”에는 각각에 대해 인간이 정답을 알려주는 레이블링(labeling) 작업이 매우 중요한 역할을 하는데 (e.g. 이 사진 속 물체는 개/고양이다.), 영상처리에 혁신을 가져왔던 대규모 사진 데이터셋, ImageNet의 기여 역시도 수천만장의 사진을 일일이 레이블링 해주는 인간의 노력이 있었기에 가능한 일이었다².

“현재까지 구글과 같은 기업에 돈을 벌어들인 기술은 컨볼루션 신경망(convolutional neural network, CNN)이나 재귀 신경망(recurrent neural network, RNN)과 같은 지도학습(supervised learning) 기술들이었다. 하지만 미래에는 비지도학습(unsupervised)이 그 자리를 대체할 것입니다.”

이는 세계 최고의 머신러닝 학회 중 하나인 NIPS(Neural Information Processing Systems)에서 딥러닝계의 스타 스탠퍼드 대학교의 앤드류 응(Andrew Ng) 교수가 남긴 말이다. 이제까지는 많은 데이터에 대한 레이블링이 지도학습의 성공을 가져왔지만, 궁극적인 데이터의 이해와 활용을 위해서는 레이블이 없는 데이터를 활용하는 비지도학습이 더욱 활발히 연구될 것이라 예측이다. 사실 이러한 예측은 인공지능 연구자라면 대부분 동의하고 있는 생각인데, 미래기술을 위해 전근대적인 반복 노동을 요구하는 지도학습은 그 과정이 현실적이지 않을뿐더러 우리가 추구하는 미래 인공지능의 모습과도 거리가 멀기에 인공지능 연구는 결국 “비지도학습”의 방향으로 나아갈 것이라 예측에 동의하는 사람이 많다.

[Contents]

Part 1. 카카오 AI 리포트 5월호 vol.3 (<https://brunch.co.kr/@kakao-it/65>)

1. Convolutional Network Models
2. Image Segmentation / Object Detection
3. Image / Video / Etc
4. Natural Language Processing / RNNs
5. Speech
6. Other Domains

Part 2. 7. Unsupervised / Generative Models
8. Understanding / Generalization / Transfer
9. Optimization / Training Techniques
10. Reinforcement Learning / Robotics
11. Epilogue

Unsupervised / Generative Models

작년을 가장 뜨겁게 달구었던 딥러닝 알고리즘을 하나 꼽으라면 단연코 GAN(generative adversarial network)[7-1]를 꼽을 수 있을 것이다. GAN³은 기존의 비지도학습들이 데이터 분포를 직접 모델링하는데 어려움을 겪었던 것을 피하기 위해 학습과정을 생성자(generator)와 구분자(discriminator)의 적대적 경쟁관계로 전환, 지도학습의 강력함을 비지도학습에 적극적으로 활용함으로써 실재와 가까운 이미지들을 생성해내는데 성공했다. 이러한 적대적 경쟁관계의 학습은 목적함수에 대한 단순 최적화(optimization)보다 까다로운 학습과정을 요구하는데, CNN과 GAN의 결합에 대한 연구[7-2]는 안정적인 학습과정을 유도하면서도 더욱 진짜 같은 이미지 생성을 가능하게 하였다.

나아가 GAN에 대한 폭넓은 실험과 학습 기술의 발전은 GAN을 준지도학습(semi-supervised learning)에도 사용할 수 있게 하였으며[7-3], 데이터의 잠재공간(latent space)에 대한 탐색도 가능하게 하는 등[7-4] 그 무궁무진한 활용가능성으로 현재까지 몇년 새 100개에 가까운 GAN 응용모델을 쏟아내었다⁴.

GAN 못지않게 각광을 받는 비지도학습 방법으로는 VAE(variational autoencoder)[7-5] 를 꼽을 수 있다. 실제 같은 예제는 생성하지만 데이터 분포의 학습에는 취약한 모습을 보이는 GAN과는 달리 VAE는 가우시안(Gaussian) 분포의 잠재공간에 대한 가정 하에 아름다운 변분 추론(variational inference)의 과정을 거쳐 제어가능한 잠재 공간을 학습할 수 있게 해주는데, 쉽게 예를 들자면, 이 잠재 공간의 레버를 조절함으로써 우리는 물체의 크기, 각도, 조도 등을 컨트롤하며 이미지를 생성할 수 있게 되었다(e.g. DRAW[7-6]). 비교하자면 GAN은 주로 진짜 같은 예제를 생성해내는 것에, VAE는 데이터의 해석과 제어에 좀더 큰 강점을 보이는데 최근에는 GAN과 VAE가 서로의 장점을 받아들이며 통합된 관점으로 해석, 발전하고 있다(e.g.⁵).

[그림 1] 구글의 BE-GAN을 통해 인공적으로 생성된 얼굴들의 예제⁶



[소개된 주요 논문들]

[7-1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).

[7-2] Radford, A., Metz, L. and Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

[7-3] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. and Chen, X., 2016. Improved techniques for training gans. In Advances in Neural Information Processing Systems (pp. 2234-2242).

[7-4] Chen, X., Duan, Y., Houthoofd, R., Schulman, J., Sutskever, I. and Abbeel, P., 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Advances in Neural Information Processing Systems (pp. 2172-2180).

[7-5] Kingma, D.P. and Welling, M., 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

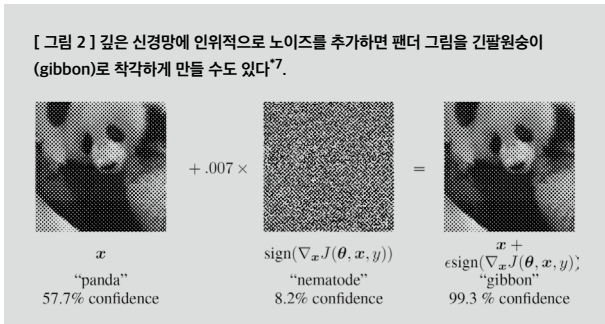
[7-6] Gregor, K., Danihelka, I., Graves, A., Rezende, D.J. and Wierstra, D., 2015. DRAW: A recurrent neural network for image generation. arXiv preprint arXiv:1502.04623.

글 | 엄태웅 terry.t.um@gmail.com

서울대 기계항공공학부에서 로봇의 모션플래닝을 전공한 뒤 회사에선 영동하게 엑소스켈레톤을 개발하다가 서른이 되어 돌연 머신러닝계로 이적한 연구계의 박쥐, 낭인, 저니맨, 능력자 친구의 친구. 현재는 캐나다 워털루공대에서 딥러닝을 이용해 휴먼모션 분석에 대한 연구(a.k.a. 틈새시장 공략)를 하고 있으며, 라임이 분유값을 벌 수 있던 피임에 솔깃해 카카오투에 글을 쓰게 되었다고 한다. 참고로 “T-Robotics” 블로그와 “테리의 딥러닝토크” 유튜브도 운영하며 최저생계비 이하의 수입을 자랑한다고 한다. 폐북에 서식하며 관심 먹고 살아가는 미물이지만, 막상 컨택하면 비싼척 하는 미청년, 아니 미야재.

Understanding / Generalization / Transfer

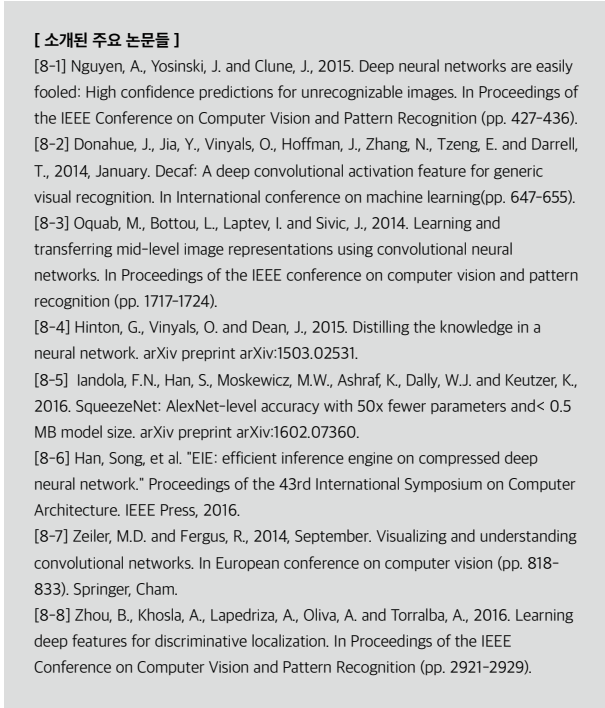
CNN, RNN의 성공부터 최근의 GAN의 돌풍까지, 딥러닝의 성공은 매우 경이적이었으나 그 성공에 대한 인간의 이해는 아직 충분치 못한 편이다. 딥러닝이 간단한 조작에 의해 쉽게 기만될 수 있다는 사실은 딥러닝에 대한 이해가 부족하다는 예 중 하나이다[8-1]. 적대적 예제(adversarial example)라 불리는 이들은 기존 이미지에 특정 레이블이 좋아하는 노이즈를 인위적으로 입힘으로써 만들 수 있는데, 이는 미래 인공지능의 취약한 보안점을 시사하기도 한다. 예를 들어 무인자동차 시대에 누군가가 교통표지판에 이러한 노이즈를 덧씌워 놓는다면 상상하기 힘든 참사로 이어질 수도 있기 때문이다. 적대적 예제의 존재는 인공지능망이 인간의 인지과정과는 다른 이해를 보여주고 있다는 증거이기도 한데, 진정 사람의 능력을 모방하는 “인공지능”을 만들고자 한다면 이 차이로부터 진정한 인공지능을 향해 의미있는 도약을 할 수 있어야 할 것이다.



딥러닝에서 더욱 깊은 이해가 필요한 또 다른 부분은 학습된 지식의 전이(transfer learning)에 대한 부분이다. 앞서 언급했던 바와 같이 모든 목표작업들에 대해 레이블링 작업을 하는 것은 쉽지 않은 일이기에, 이 경우에는 기존 지식으로부터 새로운 타겟으로의 전이(transfer) 혹은 적응(adaptation)을 통해 효율적인 학습을 달성할 수 있다. 예를 들면 개와 고양이를 구분하는데 쓰이는 학습되었던 모델을 자동차와 비행기를 구분하는 타겟 작업을 위해 재학습하는 것처럼 말이다. 신경망의 이러한 재학습 과정을 세부 조정(fine-tuning)이라 부르는데, 본 글 part 1에서 언급하였던 사물검출(object detection), 분할(segmentation)과 같은 작업들이 모두 기존 학습된 분류모델의 세부조정으로부터 출발하고 있다.

이러한 지식의 전이가 가능한 이유는 깊은 신경망이 낮은 층의 레이어(layer)들에선 타겟작업과는 독립적인, 좀더 기초적인 지식들을 학습하기 때문인데[8-2], 이러한 기초 지식들을 재활용하기 위하여 높은 층의 레이어를 재학습시키는 세부조정 방법들[8-3]이 최근 많이 활용되고 있다(e.g. 사물분류 모델을 재활용한 의료 영상분석모델⁸). 앞으로도 많은 양의 데이터 레이블이 부족한 분야에서는 비지도학습과 함께 이러한 전이학습이 적극적으로 활용될 것으로 보인다.

이 외에도 딥러닝의 혜택을 핸드폰 등의 소형기기에서도 활용할 수 있도록 거대 학습모델을 정제하여[8-4] 작은 모델에서도 동등한 성능을 낼 수 있는 방법에 대한 연구도 진행되고 있으며[8-5], [8-6] 딥러닝에 대한 분석 결과에 대한 이해를 돕기 위한 다양한 시각화 방법들도 제안되고 있다[8-7], [8-8].



Optimization / Training techniques

딥러닝의 시대 이전만 하더라도 깊은 신경망의 학습은 미분값 소멸(vanishing gradient)과 같은 현실적 문제로 인해 그 학습이 쉽지 않았었다. 이를 극복하고 많은 양의 데이터로 깊은 구조를 학습하기까지에는 ReLU(rectified unit)[9-1], [9-2]과 같은 딥러닝 구성요소에 대한 발전과 최적화(optimization) / 학습기법(training techniques)에 대한 방법론 발전이 큰 역할을 하였는데, 그 중 ADAM[9-3]은 딥러닝 학습에 가장 많이 쓰이고 있는 최적화 방법 중 하나이다. 또한 머신러닝의 고질적인 문제인 오버피팅(overfitting)을 해결하기 위해 모델의 정규화(regularization)에 대한 연구도 많이 이루어졌는데, dropout[9-4], [9-5]은 매우 간단하면서도 강력한 오버피팅 방지책을 제공해주었다.

최근 깊은 신경망의 새로운 표준으로 자리잡은 요소 중 하나는 배치분포 표준화(batch normalization)[9-6]이다. 신경망 구조는 데이터의 분포가 평균 0, 분산 1 근처일 때 가장 학습이 잘 되는 것으로 알려져 있는데, 이 때문에 데이터의 분포를 미리

조정하는 일(normalization)은 신경망 학습의 가장 기초적인 전처리과정 중 하나이다. 문제는 깊은 신경망을 통해 데이터가 변형되면서 이 분포가 무너진다는 것인데, 내재 분포 이동(internal covariate shift)이라 불리는 이 현상은 깊은 신경망의 느린 학습의 주범 중 하나였다. 배치분포 표준화는 계층마다 데이터를 다시 표준화(normalization) 해줌으로써 이 문제를 해결하였으며, 현재는 그 강력함이 대부분의 활용 모델들에서 입증되어 레이어 정규화(layer normalization)[9-7] 등 다양한 활용 방법이 존재하는 필수 레이어 구성요소로 자리잡았다.



Reinforcement learning / Robotics

강화학습(reinforcement learning)은 기존 지도학습/비지도학습의 정적인 학습방법과 달리 에이전트(agent)가 직접 환경변수들을 탐색(exploration)하며 이에 대한 보상(reward)을 확인, 최적의 행동(action)에 대한 정책(policy)을 찾아가는 생명체의 학습과 유사한 학습과정을 이용하는 방법론이다⁹. 특히 구글의 딥마인드가 이 연구영역의 강자인데, Q러닝을 이용해 아타리(Atari) 게임에서 최고점수를 기록한 인공지능의 개발이나[10-1], [10-2] 작년에 이세돌 9단을 꺾은 알파고[10-3]는 모두 이러한 강화학습의 결과물들이다. 최근 비동기식 강화학습, 일명 액터 크리틱(actor-critic)[10-4]이라 불리는 학습방식은 분산시스템을 이용해 적은 리소스로도 더욱 가볍고, 효율적이고, 안정적이게 강화학습을 할 수 있도록 해주었는데, 이는 앞으로 다수의 에이전트들이 동시에 학습을 진행해야하는 멀티에이전트(multi agent) 문제에서도 유용하게 쓰일 것으로 보인다.

로보틱스(robotics)에서도 최근 딥러닝의 성공을 적극적으로

받아들이고 있다. 특히 영상인식 분야는 이미 딥러닝이 새로운 표준으로 자리잡은 만큼 카메라 입력과 행동 결과를 end-to-end로 연결시켜 물건 집기(grasping)와 같은 작업에서 좋은 결과를 보이고 있다[10-5], [10-6]. 나아가 로봇제어의 일부분 또한 강화학습으로 대체하여 로봇 스스로 수많은 시도와 실패 끝에 물건집기를 학습하는 메커니즘을 구현하였는데[10-7], 실제 어플리케이션에선 로봇에게 수만번의 실패를 허용하기 어렵다는 점에서 앞으로 로봇들의 지식 재활용과 공유에 대한(cloud robotics) 많은 연구가 진행되어야 할 것으로 예상된다. 또한 실제 환경은 게임과 같이 행위와 보상의 관계가 명확히 규정되지 않다는 점에서 로봇의 제어를 딥러닝으로 대체하는 방향 역시 앞으로 많은 난관을 헤쳐가야 할 것으로 보인다. (참고로 로봇에서 놀라운 결과를 보여주고 있는 보스톤 다이내믹스의 로봇들의 제어에는 머신러닝이 거의 이용되지 않고 있으며, 따라서 서로 상이한 영역인 로봇제어와 딥러닝이 합쳐지려면 많은 학문적 발전이 있어야 할 것으로 보인다.)



Epilogue

지금까지 2012년부터 2016년까지 가장 많이 인용된 딥러닝 논문 100편¹⁾을 중심으로 딥러닝의 핵심 기술들과 응용분야들, 그리고 앞으로의 발전 방향까지 “겉핥아” 보았다. 2012년 AlexNet²⁾을 기점으로 폭발한 딥러닝의 급격한 성장은 CNN을 중심으로 한 비전 분야와 RNN을 중심으로 한 자연어처리, 음성인식 분야에서 성숙 단계 수준의 성과들을 보이고 있으며, 이들 영역에선 적용 도메인의 확장이나(e.g. 의료영상), 새로운 형태의 문제에 대한 도전 등(e.g. visual question answering) 끊임없는 진화를 거듭하고 있다. 이러한 지도학습 기반 딥러닝의 성공에는 ImageNet과 같은 대규모 공개데이터셋의 역할이 컸는데, 이는 다른 영역에서도 대규모 공개데이터셋이 필요함을 역설하고 있다. 따라서 우리도 각 분야에서 딥러닝 기반의 인공지능 혜택을 향유하려면 데이터 수집과 관리에 대한 정부, 기업, 대학의 역할이 더욱 강조되어야 할 것이다.

인간의 노동집약적이고 인위적인 레이블링으로부터 벗어나려는 노력도 최근 활발히 진행되고 있다. GAN과 VAE와 같은 비지도학습 방법들은 데이터 자체만을 학습하여 새로운 데이터들을 생성하는 능력을 보여주었으며, 학습된 지식의 정제와 전이는 보다 가볍고 특화 가능한 딥러닝의 활용법들을 제시하였다. 미래기술로 불리는 강화학습은 게임을 넘어 로봇을 통해 현실로 다가오고 있고, 지금도 더욱 효과적이고 안정된 딥러닝의 학습을 위해 다양한 신경망 구조와 학습기법들이 개발되고 있다. 이제까지의 딥러닝 발전이 데이터, 방법론, 연산력 혁신의 3박자가 맞아떨어져 촉발되었듯, 미래에도 각 요소 별 혁신이 병행되어야 꾸준한 인공지능 기술 향상을 이룰 수 있을 것이다.

마지막으로 이 글은 딥러닝 이해에 대한 백과사전이 아닌 딥러닝 논문을 안내하는 하나의 “메뉴판”으로서 작성되었음을 강조드린다. 각 연구의 역할을 이해하는 것과는 달리 그들을 현실에 적용하고 개선하기 위해선 훨씬 더 깊은 이해와 노력이 필요로 하므로 독자들께서는 이 메뉴판을 통해 논문 탐색의 출발점으로 삼으시어 딥러닝을 더욱 깊게 공부하고 적용하셨으면 하는 바람이다. 참고로 필자가 2015년에 쓴 “쉽게 풀어쓴 딥러닝의 거의 모든 것”이란 글³⁾이 단 1-2년 만에 옛날 이야기가 되어버렸듯, 최근 딥러닝의 연구들을 쫓은 이 글 역시도 유통기한이 그리 길지 않을 것임을 말씀드린다. 사실 2017년에 나온 논문들로만 해도 기존 몇년 간의 성과를 다시 써야할 정도인데, “최최근” 연구에 대한 소개는 다음을 기약해 보도록 하겠다.

중요한 점은 이러한 인공지능 연구의 폭발적 성장을 따라잡기 위해선 다양한 분야의 재원들과 자원들이 서로 협력해야 한다는 것이다. 그리고 그러한 협력의 바탕에는 인공지능에 대한

올바른 이해가 선행되어야 한다. 아무쪼록 우리가 그러가는 미래 인공지능이 근거없는 환상이나 막연한 기대에 의한 상상누각이 아닌, 단단한 기술 이해에 바탕한 미래 문제의 믿음직한 해결사로 자리매김할 수 있기를 희망해본다.

¹⁾ 참고 | 임태웅, “쉽게 풀어쓴 딥러닝의 거의 모든 것”, <http://t-robotics.blogspot.ca/2015/05/deep-learning.html> ²⁾ 참고 | 민현석, “알파고 만들려면 먼저 알바고 돼야”, 테크엠, http://techm.kr/bbs/board.php?bo_table=article&wr_id=3612 ³⁾ 참고 | 임태웅, “GAN, 그리고 Unsupervised Learning”, <http://t-robotics.blogspot.ca/2017/03/gan-unsupervised-learning.html> ⁴⁾ 참고 | The GAN Zoo, <https://github.com/hindupuravinash/the-gan-zoo> ⁵⁾ 참고 | Rosca, M., Lakshminarayanan, B., Warde-Farley, D. and Mohamed, S., 2017. Variational Approaches for Auto-Encoding Generative Adversarial Networks. arXiv preprint arXiv:1706.04987. ⁶⁾ 참고 | Berthelot, D., Schumm, T. and Metz, L., 2017. Began: Boundary equilibrium generative adversarial networks. arXiv preprint arXiv:1703.10717. ⁷⁾ 참고 | Goodfellow, I.J., Shlens, J. and Szegedy, C., 2014. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572. ⁸⁾ 참고 | Tajbakhsh, N., Shin, J.Y., Gurudu, S.R., Hurst, R.T., Kendall, C.B., Gotway, M.B. and Liang, J., 2016. Convolutional neural networks for medical image analysis: Full training or fine tuning?. IEEE transactions on medical imaging, 35(5), pp.1299-1312. ⁹⁾ 참고 | 최성준, 이경재, “알파고를 탄생시킨 강화학습의 비밀”, 카카오이리포트, <https://brunch.co.kr/@kakao-it/73> ¹⁰⁾ 참고 | Vinyals, O., Ewalds, T., Bartunov, S., Georgiev, P., Vezhnevets, A.S., Yeo, M., Makhzani, A., Kuttler, H., Agapiou, J., Schrittwieser, J. and Quan, J., 2017. StarCraft II: A New Challenge for Reinforcement Learning. arXiv preprint arXiv:1708.04782. ¹¹⁾ 참고 | 임태웅, “Most-cited deep learning papers”, <https://github.com/terryum/awesome-deep-learning-papers> ¹²⁾ 참고 | Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

알파고를 탄생시킨 강화학습의 비밀 part 2. (2/2)

이세돌과 알파고의 경기가 있었지 약 1년 후인 2017년 5월 당시 바둑 세계 랭킹 1위의 커제와 더욱 강력해진 알파고의 경기가 진행되었다. 알파고는 커제와 중국 기사들에게 단 한 경기도 내주지 않으며 이전에 비해 더욱 완벽해진 모습을 뽐냈다. 그리고 마치 더 이상 상대할 인간이 남지 않은 것 마냥 알파고는 커제와의 대국을 마지막으로 은퇴했다. 1년 전 이세돌이 이긴 한 경기가 인간이 알파고에서 승리한 마지막 경기로 남게 되었다.

바둑을 시작한 지 채 5년이 되지 않은 알파고가 이토록 바둑을 잘 둘 수 있었던 이유는 무엇일까? 이에 대해서 자세히 알아보기 이전에 바둑이란 게임이 왜 이제껏 컴퓨터에게 난공불락이었는지 알아보도록 하자. 일반적으로 바둑이나 체스와 같이 두 명의 선수가 번갈아가면서 하는 게임을 해결하기 위해 사용한 알고리즘은 트리 탐색(tree search) 기반의 전수 조사(발생 가능한 모든 경우의 수를 고려하는 경우)였다. 대표적인 예로 체스를 두는 IBM의 '딥블루'가 있다. 딥블루는 상대방과 자신이 둘 수 있는 경우의 수를 12수 정도를 고려하여 최적의 수를 찾으며, 당시 체스 세계챔피언이었던 러시아의 가리 카스파로프(Garry Kasparov)에게 승리를 거두었다. 당연히 딥블루 이후 인공지능과 인간의 체스 승부에서 승리의 신은 인간의 손을 들어주지 않았다.

글 | 최성준 sungjoon.choi@cpslab.snu.ac.k

서울대 전기 컴퓨터 공학부를 졸업하고, 동 대학원에서 박사 과정 중에 있다. 학부 때는 리눅스 커널을 열심히 봤었고, 회사에서 청소 로봇을 만들기도 하다가, 대학원에 와서는 기계학습과 로보틱스를 결합하는 연구를 진행하고 있다. 주로 연구하는 분야는 강화 학습과 모방 학습으로 비슷한 분야의 연구자들을 항상 찾고 있다. (연구 얘기는 언제든 환영입니다~) 주된 업무는 크게 두 가지로 나눌 수 있는데, 하나는 논문을 읽고 쓰는 것이고 다른 하나는 영수증 및 서류 처리이다. 아, 그리고 가끔씩 회사원들을 상대로 딥러닝 수업을 하기도 한다. 박사 졸업이 가까워짐에 따라 폐북에 자주 출몰하며 최근 들어 그 정점을 찍고 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 역 강화 학습을 연구하고 있다. 나를 재미있는 분야라고 생각해서 시작했는데 한국에서는 비슷한 연구를 하는 연구자를 만나기 어려운 것 같아 아쉽다. 관심 있는 사람이나 이 연구를 하고 있는 사람과의 만남이라면 언제든 환영이다. 한동안 군대 문제를 해결하기 위해 연구를 접고 영어공부에 매진했으나 영어실력은 그대로라고 한다. 최근 다시 연구를 시작하였으며 인공지능 및 로보틱스 분야의 많은 사람들과 교류하고 싶다.

체스 게임의 경우 발생 가능한 모든 경우의 수를 계산해보면 10의 100승 정도가 된다. 이 수는 엄청 큰 수임에는 분명하나 병렬 처리 기술을 활용하면 주어진 시간 내에 모두 처리 가능한 수이다. 하지만 바둑 게임의 경우 발생 가능한 경우의 수는 10의 200승이 넘어가고, 이는 전 우주에 있는 분자의 수 보다 많은 수다. 다시 말해, 현존하는 기술로는 이 모든 경우의 수를 처리할 수 없고, 당분간도 없을 예정이다.

알파고의 엄청난 기력(바둑 실력)은 크게 두 가지 기법의 조합으로 이뤄져있다. 첫 번째는 모든 가능한 수를 조사하지 않아도, 제한된 시간 내에 효과적으로 수를 계산하는 몬테카를로트리탐색(Monte Carlo tree search, MCTS)방법이고, 두 번째는 심도 학습(deep learning)을 통해서 이 MCTS를 더 효과적으로 수행하는 것이다. 여기서 심도 학습은 크게 두 가지 네트워크를 학습시키는데 사용되었는데, 현재 바둑판이 주어졌을 때, 상대방이 다음에 어디를 둘지 예측하는 policy network와 바둑의 판세를 읽는 value network이다. 심도 학습이 이러한 역할을 할 수 있었던 것은 딥신경망(deep neural network)이 다른 알고리즘에 비해 훨씬 더 많은 양의 정보를 고려할 수 있고, 이에 따른 높은 일반화의 성능 때문이다.

알파고에 사용된 MCTS 외에도 심도 학습은 일반적인 강화 학습(reinforcement learning)에도 많이 사용되고 있고, 이러한 형태의 강화 학습을 deep reinforcement learning (Deep RL)이라고 부른다. 뒤에서 더 자세히 설명하겠지만, Deep RL은 크게 두 가지로 나눌 수 있다. 첫째로는 Q function을 딥신경망으로 모델링하는 DQN(deep Q Network)와 둘째, 정책함수(policy function)를 모델링하는 심화 정책 경도(deep policy gradient)이다

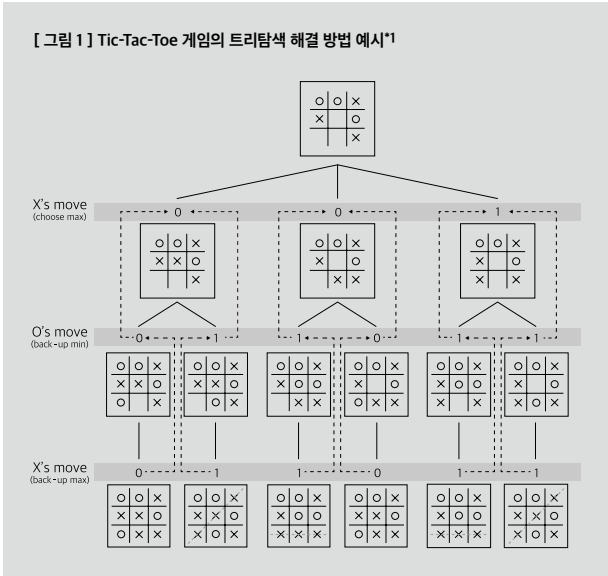
알파고의 방법론

일반적으로 기계학습에서 “Monte Calro”가 들어간 방법론들은 샘플링(sampling)을 통해서 문제를 해결하는 방법을 의미하는데, MCTS도 이와 유사하다. 체스 게임을 다루던 기존의 트리탐색 방식은 모든 경우의 수를 검색해 가장 좋은 수를 고르는 반면, MCTS는 몇 가지 해봉직한 수를 샘플링하여 검색하는 방법을 취한다. 즉 모든 수를 검색하기에는 바둑의 수가 너무 많으니 그 중에서 몇 가지 가능성이 있는 수를 우선적으로 탐색하는 것이다. 여기서 “해봉직”한 수를 고르고, 나와 상대방이 되어 수를 진행해 나가고, 현재 주어진 바둑판의 판세를 읽는데 모두 딥뉴럴네트워크가 사용된다. 강화학습에 나오는 Q value와 같은 의미로 이 수를 두었을 때 미래에 내가 얻을 보상, 즉 게임의 승패를 의미하는 값을 예측한다. 이 값을 추측하기 위해서 두 가지 값을 이용하는데 첫 번째는 승패 예측 값이다. 승패 예측을 하기 위해서

두 번째 네트워크인 value network를 이용한다. value network가 하는 일은 어떤 수를 두었을 때, 그 바둑판의 판세를 읽고 이길지 질지를 예측하는 것이다. 그리고 Q value를 추측하기 위한 두 번째 값으로 샘플링을 통한 예측 값을 사용한다. 어떤 수를 두었을 때, 그 후에 일어날 미래의 대국을 시뮬레이션 해보고 시뮬레이션 결과로부터 승패를 예측하는 것이다. 그리고 시뮬레이션 승패 결과와 value network의 승패 예측 결과를 적절히 합하여 탐색한 수에 대한 Q value를 예측한다. 이렇게 몇 가지 해봉직한 수에 대해 Q value를 예측하고 예측된 수 중 가장 좋은 수를 선택하는 것이 알파고의 알고리즘이다. 이제부터 MCTS의 원리와 MCTS에 사용되었던 네트워크(network)를 어떻게 학습시켰는지에 대해 알아보도록 한다.

트리탐색 방법(tree search)

MCTS를 알아보기 전에, 기존 방법론인 트리탐색(tree search)이 어떤 원리로 작동하는지, 어떻게 바둑처럼 번갈아가면서 플레이하는 게임을 풀 수 있는지에 대해서 살펴보도록 한다. [그림 1]과 같이 tic-tac-toe를 예시로 살펴보자.



탐색트리의 가장 위에 위치한 보드가 현재 플레이어의 상태를 의미하고, [그림 1]에서 X표시가 플레이어의 돌을 O표시가 상대방의 돌을 의미한다. 트리탐색 방법에서는, 먼저 플레이어가 할 수 있는 가능한 행동들로 첫 번째 가지를 확장시킨다. 두 번째 줄의 보드들이 플레이어가 각각의 수를 두었을 때 보드의 상태를 나타낸다. 그리고 다음 가지는 상대방이 둘 수 있는 모든 수를 고려하여 확장시킨다. 이런 식으로 플레이어와 상대방이 둘 수 있는 모든 수를 나무(tree)가지를 뒤집은 형태로 그린 것이 [그림 1]의 모습이다.

이렇게 나뭇가지를 만드는 과정을 확장(expansion)이라고 한다.

모든 수에 대해 나뭇가지를 확장하면 트리의 마지막 줄에는 게임이 끝난 상태의 보드들이 저장되고 게임의 승패 혹은 점수를 계산 할 수 있다. 이 점수를 [그림 1]의 마지막 줄에 있는 0과 1로 표현했다. 0은 무승부를 1은 플레이어의 승리를 의미한다. 이후 Tree의 마지막 줄에서부터 가장 꼭대기까지 승패의 결과를 업데이트 한다. 이 업데이트 과정을 거치면서 각 가지들은 승패에 대한 기댓값을 갖게 되고 이는 플레이어의 승률을 의미한다. 정보를 업데이트(update)한다는 것은 수를 두었을 때 플레이어가 승리 할지 패배 할지에 관해 위로 전파시키는 과정을 말한다.

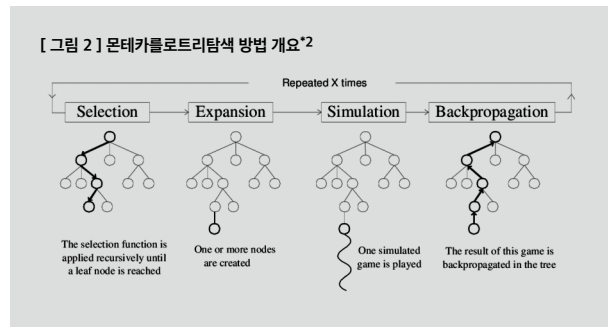
승패 정보를 밑에서부터 위로 전파시키는 데에는 따라야 할 규칙이 있는데, 이것을 최소-최대정리(Mini-Max theorem)라고 한다. 플레이어는 항상 점수가 가장 높은 수를 선택하고 상대방은 항상 플레이어의 점수가 낮아지는 수를 선택하는 것을 말한다. [그림 1]에서 두 번째 가지에서 첫 번째 가지로 승패 정보를 업데이트 시킬 때, 각각의 가지들이 두 개씩 존재하기 때문에 어느 한 정보를 선택해서 업데이트 하게 되는데, 두 번째까지는 상대방이 둘 수 있는 수를 나열 한 것이고, 상대방에 대한 업데이트 규칙이 적용된다. 즉, 더 작은 점수가 윗가지로 전파되는 것이다. 상대방은 자신의 승리를 위해서 플레이어의 점수가 최소화 되도록 행동해야하기 때문이다. 이와는 반대로 플레이어는 자신의 점수가 최대가 되는 행동을 선택하여 점수를 전파한다. 이렇게 모든 승패 정보가 트리의 뿌리(root)까지 전파되었으면, 플레이어는 승리할 확률이 가장 큰 경우의 수를 선택하면 된다. 승패를 전파하는 과정에서 상대방은 항상 플레이어가 가장 낮은 점수를 받도록 혹은 패할 확률이 높은 행동을 했기 때문에 뿌리에서는 항상 최악의 상황을 고려한 수를 두게 되어 적어도 비길 수 있게 된다.(그렇지 못하다면 그 게임은 처음부터 상대방에게 유리한 게임일 것이다.) 이런 방식을 “최악의 상황들 중 최선의 행동을 선택한다.”라고 표현 할 수 있고 이 원리를 최소-최대정리라고 하는 것이다.

트리탐색 방법을 간단하게 요약하면, 상대방의 수를 모두 고려하여 트리를 만든 뒤 트리의 마지막 노드(node)를 이용하여 게임의 승패를 결정하고, 이 정보를 트리의 뿌리까지 전파(정보 업데이트)시키는 것이다.

몬테카를로트리탐색 방법

바둑 게임에 트리 탐색기법을 적용한다고 생각해보자. 첫수부터 시작하여 모든 경우의 수를 트리로 만든다면 엄청난 수의 가지치기를 해야 할 것이다. 만약 이것이 가능하다면 바둑의 필승 법 혹은 지지 않는 법 등이 개발되었을지도 모른다. 그러나 바둑의

모든 수를 나타내는 방대한 트리를 저장할 메모리와 이 트리의 승패 정보를 처리할 수 있는 컴퓨터는 아직까지 존재하지 않는다. 이런 문제를 해결하기 위해서 트리탐색에 몬테카를로 방법을 적용하게 된다. 몬테카를로 방법은 대부분의 경우 샘플링 기법이라고 바뀌서 말해도 의미가 통한다. MCTS는 기존의 트리탐색과 같이 모든 경우의 수를 조사하는 것이 아니고 몇 가지의 경우를 샘플링하여 조사하는 것이다. 샘플링에 기반을 둔 MCTS 알고리즘은 [그림 2]와 같은 순서로 진행된다. 선택(selection), 확장(expansion), 시뮬레이션(simulation), 역전파(back propagation) 네 단계를 거치면서 문제의 답을 찾아가게 된다.



MCTS의 작동 과정을 간단히 설명하면, 먼저 어떤 수를 탐색할지 선택(selection)하고, 탐색하고자 하는 경우의 수로 확장(expansion)한 뒤, 확장한 수의 승패를 예측(simulation)하고, 그 결과를 나무의 뿌리쪽으로 업데이트 하는 역전파(back propagation)과정으로 이루어진다. 뒤에서 자세히 설명하겠지만, 알파고에서는 선택 과정과 예측 과정에서 컨볼루션 뉴럴 네트워크(convolutional neural network)가 사용되었다.

[그림 3]은 MCTS 네가지 방법이 반복(iteration)적으로 이루어지는 모습을 나타낸 것이다. 각각의 방법을 자세히 알아 보도록 하자. 먼저, 선택 과정에서 어떤 노드, 어떤 수를 탐색할지 선택하게 된다([그림 3]의 오각형 모양 노드). 이때는 특정 규칙을 따라서 탐색할 경우의 수를 선택하게 되는데, 이런 규칙을 선택 규칙(selection rule)이라고 한다. 알파고에서는 UCT(upper confidence bounds for trees)라는 방식의 선택 규칙을 사용하였다. 알파고에 적용된 UCT를 설명하기 전에 이러한 선택 규칙이 MCTS의 성능에 미치는 영향을 먼저 생각해보자. 어떤 경우의 수를 탐색할지 선택하기 위해서 주로 두 가지 기준을 이용한다. 첫 번째는 ‘얼마나 다양한 수를 탐색할지’ 그리고 두 번째는 ‘한 가지 수를 얼마나 여러 번 시뮬레이션 할지’이다. 기계학습(machine learning)에서는 이런 기준을 탐색과 활용(exploration and exploitation)라고 부른다. 경우의 수를 다양하게 탐색하는 것은 어떤 식으로 성능에 영향을 미치게 될까? 직관적으로 생각해보면, 좋은 수를 시뮬레이션 해보지 못하면 좋은

수의 승률을 예측하지 못하고 최종적으로는 다음 경우의 수를 선택 하지 못하는 것이다. 따라서 다양한 수를 탐색하는 것은 좋은 수를 찾아내는데 있어서 중요한 역할을 한다. 두 번째로 한 가지 경우의 수를 여러 번 시뮬레이션 하는 것은 탐색하는 수의 승률 예측의 정확도를 결정하게 된다. 이 시뮬레이션 과정이 몬테카를로 또는 샘플링 과정이기 때문에 많은 샘플을 뽑으면 뽑을수록 높은 정확도로 승률을 예측하게 된다. 정확한 승률 예측을 해야 좋은 경우의 수와 나쁜 경우의 수를 정확히 구별해 낼 수 있게 된다. 선택(selection) 과정에서는 매번 새로운 수를 시뮬레이션 할지 아니면 이전에 탐색했던 수를 다시 시뮬레이션 하여 승률 예측의 정확도를 높일지를 비교하여 탐색할 수를 결정 하게 된다. 이때 탐색(exploration)을 너무 많이 하게 되면 승률 예측의 정확도가 떨어지게 되고 반대로 활용(exploitation)을 너무 많이 하게 되면 다양한 경우의 수를 탐색하지 못해 최적의 수를 찾지 못하게 된다. 따라서 이 둘 사이의 적절관계(tradeoff)를 고려하여 탐색과 활용을 하는 것이 MCTS의 핵심이다. 추가적으로 tradeoff는 강화학습 에서도 중요하게 다뤄지는 문제이다.

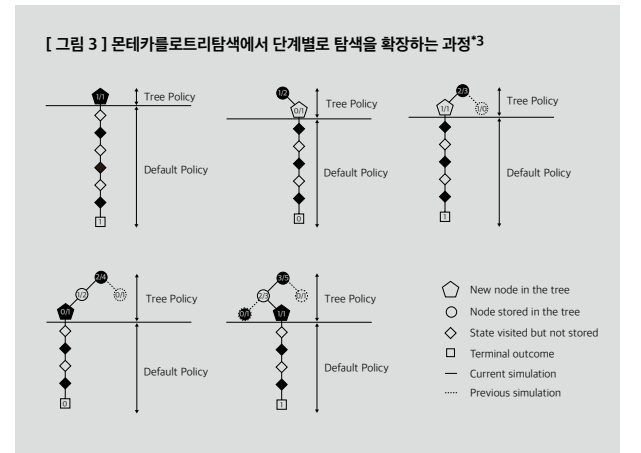
알파고에서는 이러한 exploration and exploitation tradeoff를 해결하기 위해서 아래와 같은 UCT 형태의 선택 규칙을 사용한다.

$$a_t = \operatorname{argmax}_a (Q(s_t, a) + u(s_t, a)), u(s, a) \propto \frac{P(s,a)}{1+N(s,a)}$$

st 는 현재 주어진 바둑판을 의미하고, at 는 다음에 탐색해야할 수(행동)를 의미한다. 그리고 Q(st, a) 는 지금까지 예측한 각각의 경우의 수 a에 대한 승률을 나타내고 u(st, a) 는 보너스 값으로 P(s,a)와 N(s,a) 에 의해서 결정된다. P(s,a) 는 프로기사들의 기보를 통해서 학습하는 값으로 프로기사들은 현재의 바둑판 상황 st 에서 a라는 수를 선택할 확률을 의미한다. 그리고 N(s,a)은 a라는 수가 탐색 과정에서 얼마나 선택 되었는지를 말한다. 따라서 u(st,a) 는 a라는 수가 많이 선택되게 되면 N(s,a)이 증가하여 줄어든게 되는 값이고 P(s,a)에 비례하므로 프로기사들이 둘 확률이 높은 수에 대해서는 큰 값을, 둘 확률이 낮은 수에 대해서는 작은 값을 갖게 되어있다. u(st, a)의 역할은 프로기사들이라면 선택할 만한 다양한 수들이 탐색하도록 만드는 것이다. Q(st, a) 와 u(st,a) 값이 경쟁하면서 탐색과 활용을 조절하게 된다. MCTS의 초반에는 시뮬레이션 횟수 N(s,a)가 작기 때문에 u(st,a)가 Q(st,a)에 비해서 더 큰 값을 갖게 되고 따라서 다양한 수를 탐색하게 된다. 그리고 많은 시뮬레이션을 거쳐 N(s,a)가 적당히 커지게 되면 u(st,a)가 작아지면서 Q(st, a)의 영향을 받게 되고, 승률이 높은 수를 집중적으로 탐색하게 된다. 이러한 과정에서 가능한 모든 수를 동등하게 탐색하지 않고 P(s, a)에 비례하여 탐색하기 때문에

프로기사들이 주로 두었던 수에 기반하여 탐색을 하게 된다. 그리고 P(s,a)를 모델링 하기 위해서 딥 컨볼루션 뉴럴 네트워크(deep convolutional neural network)가 사용되었다.

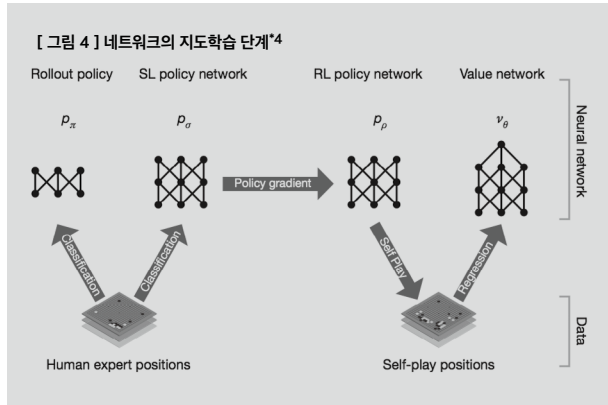
한 가지 수를 탐색한다는 것은 시뮬레이션을 통해 게임을 끝까지 진행하고 그 시뮬레이션의 승패 결과를 얻는 것을 말한다. 시뮬레이션하기 위해서는 플레이어와 상대가 어떻게 행동할 것인가, 혹은 바둑으로 치자면 어떤 수를 둘 것인가에 대한 행동 양식이 있어야 한다. 이것을 정책(policy)이라고 부르는데 [그림 3]에서는 시뮬레이션에 사용되는 정책을 기본정책(default policy)라고 표시하였다. 가장 쉽게 적용 할 수 있는 방법은 기본 정책으로 랜덤하게 행동을 취하는 무작위정책(random policy)를 사용하는 것이다. 다시 말해 무작위로 수를 두는 것이다. 또는 시뮬레이션의 정확도를 높이기 위해서 다양한 휴리스틱(heuristic)들을 사용한다. 알파고에서는 기본 정책에 프로기사들의 대국 기보를 기반으로 학습한 rollout policy network를 사용함으로써 시뮬레이션의 정확도를 높였다. 그리고 이 시뮬레이션 결과를 보완 해주는 역할로 value network를 사용하였다. Value network는 시뮬레이션을 하지 않고, 현재 주어진 바둑판을 보고 승패를 예측하는 역할을 하여서 시뮬레이션 횟수를 줄이는데 도움을 주었다. 그리고 이 value network의 승패 예측이 꽤 정확히 작동했기 때문에 알파고가 좋은 수를 찾을 수 있었다. 시뮬레이션이 끝나면 승패 결과를 얻게 되는데 [그림 4]에서 네모 상자에 써져있는 것이 승패의 결과이다. 1은 승리 0은 패배를 의미한다. 그 뒤에는 승패 결과를 탐색을 시작한 오각형 모양의 노드에 업데이트 한다. 따라서 탐색을 반복하면 반복 할수록 승패 결과가 쌓이게 되고 이 값을 통해 각각의 수들의 승률을 추정 할 수 있게 된다. [그림 3]의 각 노드에 써져있는 숫자들이 바로 승률을 의미한다. 그리고 충분한 횟수만큼 탐색을 한 뒤에는 승률이 가장 높은 수를 선택하는 것이 MCTS의 작동 방식이다.



value network 와 policy network

지금부터는 딥뉴럴네트워크가 MCTS에서 어떻게 활용되고 있는지 알아보도록 하자. 먼저, 각각의 네트워크의 역할과 학습 방식을 설명하고 그 네트워크들이 MCTS의 어느 부분에 들어가게 되는지 설명하려고 한다. 알파고는 총 세 가지 뉴럴 네트워크(neural network)들을 사용하였다. 각각 뉴럴 네트워크의 이름은 rollout network, supervised policy network, 그리고 value network이다. 앞의 두 가지 네트워크는 현재 바둑판의 상태를 파악하고 다음 수를 예측하는 역할을 하고, value network는 현재 바둑판의 상태를 파악해서 게임의 승패를 예측하는 역할을 한다.

이 세 가지 네트워크들의 공통점은 모두 '예측'하는 역할을 한다는 것이다. 다음 수를 예측하거나 승패를 예측하는 것을 머신러닝의 분류(classification) 문제로 바꿀 수 있는데, 알파고는 바로 이 분류 문제에 딥러닝을 적용함으로써 큰 성능 향상을 가져올 수 있었다. 다음 수를 예측하는 것은 현재 바둑판에 놓여있는 바둑알을 입력하여 다음 19x19가지의 경우의 수중에 한 가지를 선택하는 문제로 볼 수 있다. 그리고 승패를 예측하는 것은 현재 바둑판 상태를 입력해서 승 혹은 패, 둘 중 한 가지를 선택하는 문제로 볼 수 있다. 이렇게 몇 가지의 선택지 중 한 가지를 고르는 분류 문제에서 딥러닝이 탁월한 성능을 보여주고 있다. 최근들어 다양한 딥러닝 기법들이 개발되면서 분류 문제에서 딥러닝이 탁월한 성능을 보여주었고 알파고는 이를 바둑에 적용한 것이다. 예를 들어 이미지를 보고 이미지 속에 어떤 물체가 있는지 고르는 문제, 혹은 사람이 있는지 없는지, 입력한 이미지가 고양이인지 아닌지 등을 맞추는 것은 딥러닝이 탁월한 성능을 보이는 분류 문제들이다. 바둑판의 판세 예측, 다음에 둘 수의 예측 문제들을 분류 문제로 바라보고 딥러닝을 적용한 것은 딥러닝이 좋은 성능을 보일 수 있는 문제에 적절히 적용한 것이라고 생각한다.

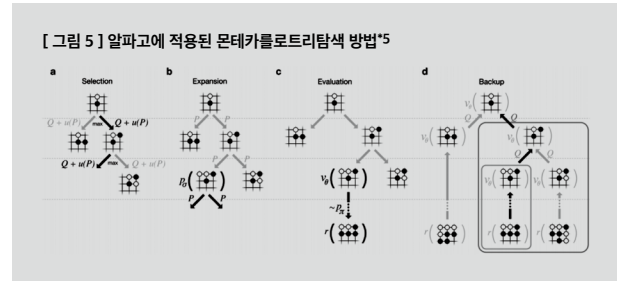


이제부터 rollout policy, supervised policy, value network를 살펴보도록 하자. Rollout policy와 supervised policy는 같은 것을 학습 한다. 현재 바둑판 상태를 파악해 다음 수를 어디에

두어야 하는지를 예측하는 것인데, 왜 두 가지 네트워크를 같이 학습시킬까? 이유는 서로 같은 것을 학습하지만 쓰이는 곳이 다르기 때문이다. Rollout policy는 MCTS의 시뮬레이션에 사용되고 supervised policy는 선택 과정에 사용되어진다. 이 때문에 학습하는 것은 같지만, 그 구조에 차이가 있게 된다. Rollout policy는 히든 레이어(hidden layer)가 없는 아주 간단한 형태의 구조를 사용하였고 supervised policy는 히든 레이어가 11층 있는 복잡한 구조를 사용하였다 [3]. Rollout policy를 간단한 형태로 만든 이유는 시뮬레이션 속도를 빠르게하기 위해서다. 딥뉴럴네트워크의 특성상 히든 레이어가 많아지면 변수(parameter)가 늘어나고 연산 횟수가 늘어나게 된다. 실제로 몇 만 번의 시뮬레이션을 하는 MCTS에서 히든 레이어가 많은 복잡한 형태의 네트워크를 사용하게 되면 빠른 계산이 어렵게 될 것이고 이를 피하기 위해서 rollout policy는 매우 간단한 형태의 구조를 사용하였다. 반대로 MCTS의 선택 과정에 사용되는 supervised policy는 상대적으로 복잡한 구조의 네트워크를 이용하였다. Supervised policy는 앞서 나온 수식으로 학습을 하게 되는데, 이에 활용한 데이터가 프로기사들의 기보들로서 프로기사들이 주로 두는 경우의 수들을 정답(supervised)의 확률로 기억하게 된다. 즉, 프로기사들이 주로 뒀던 수들은 높은 확률을 갖도록 학습된다. 즉 탐색 과정에서 의미 없는 수를 탐색하는 것이 아니라, 프로기사들이 주로 두는 수를 우선순위로 하여 탐색을 진행 하게 해주는 역할을 한다. 이 네트워크가 rollout policy에 비해 복잡한 이유는 시뮬레이션에 사용되지 않아 상대적으로 느리게 학습이 진행되고, 높은 정확성을 필요로 하기 때문이다. 일반적으로 네트워크의 깊이(히든 레이어의 개수)가 많아지면 성능이 높아진다고 알려져 있기 때문에 많은 양의 기보를 더 정확히 학습하기 위해서 supervised policy는 rollout policy에 비해 더 복잡한 네트워크를 사용하게 된 것이다.

마지막으로, value network는 현재 바둑판의 상태를 통해 승패를 예측하도록 학습되었는데, 이 네트워크를 통해서 시뮬레이션 횟수를 줄일 수 있었다고 본다. 기존 강화 학습에 많이 사용하던 rollout policy의 시뮬레이션에 의존한 승패 예측 방식이 알파고에서는 시뮬레이션과 value network를 같이 사용하면서 많은 시뮬레이션을 하지 않고 기보 데이터의 도움을 받아 적은 시뮬레이션으로도 높은 승패 예측 정확도를 낼 수 있었다고 보는 것이다. 이때 value network를 학습시키기 위해서 기존에 한국의 프로기사들이 두었던 대국의 기보와 알파고 끼리 둔 바둑 기보를 합하여 학습을 하게 했는데, 이 부분에서 재미있는 사실을 한 가지 발견 할 수 있다. 알파고에 사용한 value network는 왜 프로기사들의 기보와 알파고 끼리 대국한 기보를 합쳐서 학습 한 것일까? 바로 프로기사의 기보에 있는 바이어스 효과를 없애기

위해서다. 프로기사들에게는 보통 기풍이라는 것이 존재 한다. 예를 들면, 이창호와 같은 기사는 치밀한 계산 하에 수비적인 바둑을 구사하는 반면, 이세돌 같은 바둑기사는 꽤 공격적으로 바둑을 둔다. 기풍이라는 것은 프로기사의 기보 데이터가 다양한 샘플을 담고있지 못하게 하는 것이다. 즉, 기보를 만들어내는 기사들은 저마다의 기풍으로 바둑을 두기 때문에 다양한 기보가 생성되는 것이 아니고, 비슷비슷한 형태의 바둑기보들이 생성되는 것이다. 여기서 생기는 바이어스 때문에 실제로 프로기사의 데이터를 이용하여 학습한 경우 학습에 사용된 트레이닝 셋에서는 높은 예측 정확도를 보이지만 학습에 사용되지 않은 테스트 셋에서는 낮은 예측 정확도를 보인다. 이러한 경우를 일반화(generalization)가 좋지 못하다고 하는 것으로, 학습에 사용된 데이터에 대해서는 높은 확률로 승패 결과를 맞추지만 학습에 사용되지 않은, 즉 처음 보는 데이터에 대해서는 승패 결과를 잘 맞추지 못하는 것을 말한다. 이런 현상을 알파고에서는 알파고 끼리 대국한 기보 데이터를 추가하여 학습함으로써 해결하였다. 여태까지 설명한 세 가지 네트워크들은 [그림 5]와 같이 선택과 시뮬레이션(simulation, 다른 말로는 evaluation)에 사용되고 있다.



종합하자면, 알파고의 알고리즘은 딥러닝을 기존의 방법론인 MCTS의 선택과 시뮬레이션 과정에 적용한 것이다. 승패 예측과 탐색의 우선순위 등을 많은 양의 데이터를 통해 학습된 딥러닝과 기존의 방법론인 MCTS를 융합하여 극복하였다는 사실은, 딥러닝을 다른 분야에 적용하는데 있어서 본받을 만한 선례를 만들었다고 생각한다.

DQN과 DQN의 학습 기법

이제 다시 강화학습으로 돌아와서 강화학습에서 딥러닝이 사용된 배경과 그 학습 방법을 알아보자. 딥강화학습(deep reinforcement learning)은 벽돌깨기와 같은 아타리(Atari) 게임 문제를 해결한 deep Q learning에서 시작하여 지금은 연속공간에서의 정책함수(policy function)와 그 정책함수의 좋고 나쁨을 평가하는 Q함수(Q function)를 동시에 학습하는 actor critic 방법까지 진화 하였다. 강화학습 입장에서 딥러닝은 Q함수를 모델링하기

위한 함수 추정 모델 중 한 가지 불과 하다. 일반적인 regression 기법들을 강화학습에 적용 시킬 수 있도록 이론적으로 발전해 왔다. 딥러닝 이전에는 커널 회귀 분석(kernel regression)을 이용한 방법들이 많이 연구되었다. 그러나 딥러닝 알고리즘 발전과 GPU의 등장에 힘입어 커널 방법을 제치고 deep Q learning이 강화 학습에 사용되고 있다. 최근 딥강화학습 연구 결과들은 이미 과거에 간단한 함수 추정 방법론을 이용하여 연구되었던 방식들을 딥뉴럴네트워크로 바꾼 경우가 많이 있다.

강화학습과 DQN의 함수 추정(function approximation)

강화학습에서 함수 추정(function approximation)이 적용된 이유는 무엇일까? 함수 추정의 특성은 일반화(generalization)에 있는데 바로 이 일반화 특성을 이용하여서 학습한 Q 함수는 모델에 효율적으로 사용될 수 있게 한다. 강화학습이 진행 되는 과정은 간략히 두 과정으로 나뉘는데 샘플링을 통해 에피소드들을 모으고 모여진 에피소드들을 기반으로 Q 값과 정책값을 업데이트 시키는 것이다. 강화학습이 이러한 샘플링 기법이기 때문에 발생하는 한 가지 문제점이 있는데 바로 차원의 저주(curse of dimensionality)이다. 일반적으로 상태 공간의 크기가 커지면 커질수록 Q 값이 수렴하는데 더 많은 에피소드들을 필요로 하게 된다. 예를 들어 9*9 바둑 문제를 강화 학습을 통해서 풀려고 하면, 전체 상태 공간이 10^{38} 정도로 어마어마하게 큰 공간에서 샘플링을 해야 한다. 따라서 수렴속도가 현저하게 떨어지게 된다.

이러한 문제를 해결하기 위해서 함수 추정 기법을 사용하게 되는 것이다. 현재 상태를 잘 표현해 줄 수 있는 특징(feature)들을 추출하고 이 특징에 대한 Q 함수를 학습시키게 된다. 이런 방식을 취하는 이유는 일반화 효과를 이용하려는 것이다. 샘플링이 많이 되지 않은 지역일지라도 Q 함수의 일반화가 잘 되어있다면, 처음 마주 하는 상황에서도 정확한 Q 값을 예측 할 수 있을 것이다.

이렇게 특징을 이용하는 것은 문제를 기술하는 표현(Representation)을 변경하는 것이다. 다시 9*9 바둑의 예를 들면, 고전적인 강화 학습 기법을 이용하기 위해서는 바둑판의 가능한 모든 경우의 수에 대해서 Q 값을 찾아야 한다. 10^{38} 개의 상태가 존재 하고 각 상태에서 취할 수 있는 행동까지 고려한다면, 어마어마하게 많은 Q 값을 찾아 내야 하는 것이다. 그러나 만약, 우리가 바둑 판을 몇 가지 중요한 특징을 이용하여 나타내면 적은 샘플로 여러 상태의 Q 값을 추론 할 수 있게 된다. 예를 들어 바둑판을 보고 현재 상대방의 집 수와 자신의 집 수, 혹은 '패'가 가능한지, '축'이 가능한지의 여부로 현재의 바둑판을 나타낸다고 해보자. 이러한 특징들을 바둑판으로부터 추출하면, 처음보는 수,

처음보는 바둑판일지라도 비슷한 특징을 갖고 있다면 현재 상태가 좋은지 나쁜지, 어떤 행동을 해야 하는지를 추론 할 수 있게 된다.

Q 함수를 특징을 이용한 함수 추정 기법으로 학습시키기 위해서는 수식 상 약간의 재정의가 필요하다. 기존의 강화학습에서 사용되었던 업데이트 식을 다시 한번 살펴보면 아래와 같다.

$$Q(s, a) \leftarrow R(s, a) + \gamma \max_{a'} Q(s', a')$$

위 수식에서 오른쪽 항을 타겟값(target value)이라고 한다. 함수 추정 기법을 이용하지 않은 일반적인 강화 학습에서는 모든 상태와 행동 쌍에 대하여 이 타겟값을 업데이트하면 되지만, 함수 추정 기법을 이용하면 우리가 업데이트해야 할 것이 함수의 파라미터로 바뀌게 된다. 기존의 모든 상태와 행동에 대해 정의 됐던 Q 함수를 아래와 같이 근사 하는 것이다.

$$Q(s, a) \approx \hat{Q}(\phi(s), a : \theta)$$

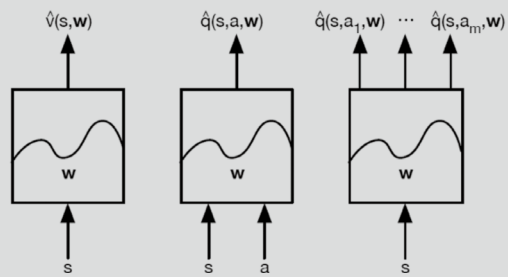
이렇게 Q 값을 추정 하게 되면, 가보지 못한 상태(state) 라고 해도 비슷한 특징 점을 갖는 상태라면 Q 값을 추론 할 수 있다는 장점이 있다. Q 네트워크를 업데이트하는 방식은 다음과 같은 손실(loss)을 최소화 하는 파라미터를 찾는 것이다.

$$L(\theta) = (y - \hat{Q}(\phi(s), a : \theta))^2$$

$$y = R(s, a) + \gamma \max_{a'} \hat{Q}(\phi(s'), a' : \theta)$$

이 손실 함수의 의미는 새로운 타겟값에 가장 가까운 예측을 하는 파라미터를 찾는 것이다. Q 를 추론하는 네트워크를 디자인 하는 방법은 여러 가지가 있는데 가장 많이 사용하는 것은 [그림 6]의 세 번째 방식이다. 뉴럴네트워크의 입력은 상태(state)가 되는 것이고, 출력은 각각의 행동을 취했을 때 얻을 수 있는 Q 값을 유추 한다. 참고로 알파고 에서는 첫 번째 형태의 네트워크가 사용되었다.

[그림 6] Q함수의 파라미터(parameter)를 찾는 방식



이러한 형태의 디자인을 위해서는 한 가지 조건이 필요한데 바로 취할 수 있는 행동의 개수가 유한해야 한다는 것이다. 예를 들어 바둑 게임은 경우의 수가 많긴 하지만 행동의 최대 개수는 19*19로

제한 되어있다. 반면에, 자율 주행을 하는 자동차를 학습시키는 문제의 경우에는, 액셀을 어느 정도 밟을지, 핸들을 어느 각도로 틀지와 같이 연속적인 값을 갖기 때문에 유한한 행동으로 표현하기 힘들다. 이렇게 연속적인 행동 공간에서 deep Q learning을 적용하기 위해서는 연속공간을 이산화(discretization) 해야 한다. 하지만 연속적인 행동공간을 이산화 시키지 않아도 적용 할 수 있는 방법이 있는데 이것이 바로 policy gradient과 actor critic 모델이다.

Double DQN 과 듀얼 네트워크(dueling network)

DQN이 네이처(Nature)지에 실린 이후에 많은 연구자들이 deep Q learning을 연구하기 시작했다. 첫 번째 소개할 방식은 학습방식과 네트워크 구조에 관한 것이다. DQN을 학습시키다보면 네트워크 성능이 불안정하게 증가하는 것을 볼 수 있다. 과하게 좋은 Q 값을 주어서 네트워크가 더 이상 성능 향상을 하지 못하는 경우가 발생한다. 이런 현상들은 딥러닝 방법을 도입하기 이전 부터 발생하던 현상으로, 일반적인 함수 추정을 이용해도 주로 나타나는 현상이었다. 이 현상을 막기 위해서 등장한 기법이 double DQN이다. 이름에서도 알 수 있듯이, double DQN은 두개의 네트워크로 구성되어 있다. 이 두 개의 네트워크를 이용해서 안정적이고 정확한 Q 값을 학습을 할 수 있다. 두 개의 네트워크 중 한 네트워크는 빠르게 업데이트하면서 에피소드를 만들어 내는 역할을 하고 다른 네트워크는 상대적으로 느리게 업데이트 하면서 어느 한 행동의 Q 값을 과도하게 커지지 않도록 해준다. 또한, 기존의 DQN에 비해 Q 값이 천천히 변하기 때문에 훨씬 안정적인 학습이 가능하다. Double DQN과 DQN의 차이는 타겟값을 구하는 방식에 있다. 아래와 함수가 Double DQN의 학습에 사용된다.

$$y = R(s, a) + \gamma \hat{Q}(\phi(s'), a' : \theta^-)$$

$$a' = \operatorname{argmax}_a \hat{Q}(\phi(s'), a : \theta)$$

위의 수식에서 θ^- 와 θ 는 double DQN에 있는 두 네트워크의 파라미터를 의미한다. 하나는 느리게 업데이트 되는 네트워크를 의미하고, 다른 하나는 빠르게 업데이트 되는 네트워크를 의미한다. 먼저 θ^- 를 업데이트 하는 방식은 간단한데, 일정한 반복횟수 마다 업데이트를 실시해 주게 된다. 이와 같은 방식으로 타겟값을 구한다는 것은 현재까지 추정된 Q 값 중 가장 좋은 행동을 고르고 이 행동에 대한 타겟값을 구할 때는 기존 업데이트된 파라미터를 추정된 Q 값에 이용하는 것이다. 기존의 방식에서는 항상 Q의 최댓값을 이용하여 업데이트가 진행되었기 때문에 한 번 잘못 Q 값을 추정하기 시작하면 그 에러가 계속해서 커지는 형태였다면,

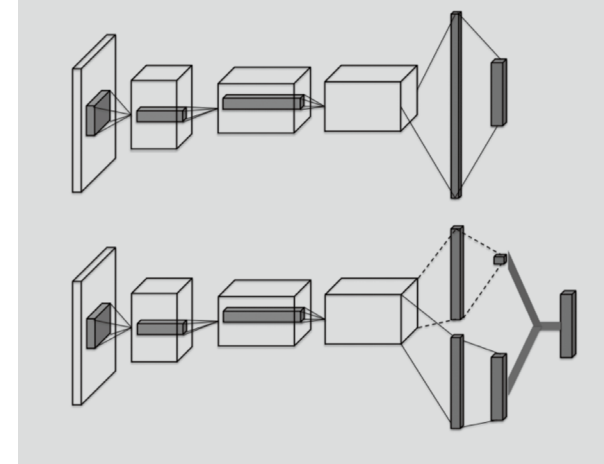
double DQN에서는 네트워크 파라미터가 천천히 업데이트되기 때문에 에러의 전파가 빠르지 않고 안정적인 학습이 가능해진다.

두 번째로 소개할 방식은 듀얼 네트워크(dueling network) 방식이다. [그림 7]과 같은 새로운 DQN 구조를 제안했다. Q 값의 정의로부터 유도되는 한 가지 수식에서 아이디어를 얻은 새로운 DQN구조를 가지고 있다.

$$Q(s, a) = V(s) + A(s, a)$$

Q 값의 의미는 현재 상태에서 행동을 취할 때 얻을 수 있는 보상의 합을 의미한다. 이 값을 두 가지로 분리해서 생각해보면 벨류(V)와 어드벤처(A)가 된다. V 값이 의미하는 것은 현재 상태에서 최선의 행동을 취했을 때 얻을 수 있는 보상의 합이다. 그리고 A는 최선인 행동과 다른 행동들 사이의 보상의 차를 의미한다. 듀얼 네트워크는 기존의 Q 값을 V와 A로 나누어서 예측하는 [그림 7]과 같은 구조를 제시하였다. 이러한 구조를 제시하는 이유는 간단한 직관으로부터 나왔다고한다. Q 값을 추론하는 것을 두 가지로 분리해서 생각한 것인데, 현재 상태가 좋은지 나쁜지를 V 값으로 추론하고 그 중에서 어떤 행동을 고를지를 A 값을 이용하여 추론하였다. 즉, V 값은 바이어스 같은 역할을 하고 V를 중심으로 좋고 나쁨을 A 값을 이용하여 추론 하게 되는 것이다.

[그림 7] 듀얼 네트워크 구조 (dueling network)*6



*1 출처 | <http://snipd.net/minimax-algorithm-with-alpha-beta-pruning-in-c>. *2 출처 | https://www.researchgate.net/publication/23751563_Progressive_Strategies_for_Monte-Carlo_Tree_Search *3 출처 | <https://jay.tech.blog/2017/01/01/heuristic-search-mcts/> *4 출처 | <https://blog.acolyer.org/2016/09/20/mastering-the-game-of-go-with-deep-neural-networks-and-tree-search/> *5 출처 | <https://blog.acolyer.org/2016/09/20/mastering-the-game-of-go-with-deep-neural-networks-and-tree-search/> *6 출처 | Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, Nando de Freitas: Dueling Network Architectures for Deep Reinforcement Learning. ICML 2016: 1995-2003
 【참고 자료】 · Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., and Petersen, S. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533. · Van Hasselt, Hado, Arthur Guez, and David Silver. "Deep Reinforcement Learning with Double Q-Learning." AAIL, 2016. · Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., & Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484-489.

앤드류 응의 코세라 딥러닝 전문가 과정 소개

지난 3월 스탠퍼드대 앤드류 응교수는 바이두(Baidu)를 떠나면서, 전세계적인 AI 커뮤니티를 위해 지원을 계속하겠다고 밝혔다. 많은 이들이 응 교수의 다음 행보를 궁금해 했는데, 그것이 밝혀지기엔 오래 걸리지 않았다. 지난 6월 응 교수는 deeplearning.ai 프로젝트를 시작했다고 밝혔다. AI 연구자들 혹은 잠재적 연구자들을 위해 온라인을 통해 좋은 교육 콘텐츠를 제공하는 것이다. 이번 8월에 응 교수가 설립에 참여한 무료 온라인 교육 사이트인 코세라(Coursera)를 통해서 5개의 딥러닝 전문가 과정을 오픈했다. 이번 호에서는 이 과정들을 소개하려고 한다. 현재 수강할 수 있는 과정은 5개 중 3개로 나머지 2개의 과정도 순차적으로 제공될 예정이다. 모든 강의는 영어로 제공되고 있으며, 한글 자막은 없다. 이번 응 교수의 강의는 딥러닝을 체계적으로 공부해보고 싶은 사람들에게는 유용한 참고자료가 될 것으로 평가받고 있다.



1. Neural Networks and Deep Learning

최근 AI연구의 꽃을 피우고 있는 딥러닝에 대해 체계적으로 배울 수 있다. 뉴럴네트워크(neural network) 구조, 이를 활용한 분류(classification) 방법, 알은 뉴럴네트워크 구성 및 문제에 적용하는 방법, 효과적으로 뉴럴네트워크를 구성하는 방법, 딥러닝 설계를 위한 중요한 파라미터를 이해할 수 있다.

기간 | 4주, 주당 3~6시간 소요

2. Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization

딥러닝이 해결하고자 하는 문제에 잘 작동시키기 위한 응용 방법을 주로 학습하게 된다. 응 교수는 오랜기간 딥러닝을 연구해온 경험을 바탕으로 딥러닝이 문제를 해결해나가는 과정을 알기 쉽게 설명해주고 있다. 또한, 알고리즘을 설계하면서 경험적으로 해결해야 하는 것들이 어떤 것들인지 알려주고 있다. 구글에서 제공하고 있는 머신러닝 프레임워크인 텐서플로우를 활용한 실습이 진행된다.

기간 | 3주, 주당 3~6시간 소요

3. Structuring Machine Learning Projects

성공적인 머신러닝 프로젝트를 만들기 위한 방법을 학습하게 된다. 응 교수가 진행한 다양한 머신러닝 프로젝트의 경험을 강의에 녹이고 있다. 응 교수는 여러 곳의 강연을 통해 딥러닝과 시가 다양한 분야에서 활용되기를 기대하고 있고 그렇게 되기 위해 노력하고 있다. 강의에서는 실제 프로젝트 실습을 통해서 어떻게 문제에 대처하고, 의사결정을 도와줄 수 있는지 배우게 된다.

기간 | 2주, 주당 3~4시간 소요

4. Convolutional Neural Networks

아직 공개되지 않은 2개의 강의 중 하나이다. CNN은 최근 컴퓨터비전(computer vision, 컴퓨터를 통한 시각이미지, 영상 처리를 연구하는 분야)에서 높은 성과를 보여주고 있는 알고리즘이다. 코스 소개에는 CNN의 구조를 이해하고, 다양한 변화를 통해 시각이미지를 인식하는 알고리즘을 구현하는 걸 주로 다루고 있다. 이미지 뿐 아니라 비디오, 최근에 많이 활용되고 있는 3D영상 데이터까지 실습해 볼 수 있을 것으로 보인다.

기간 | 미정

5. Sequence Models

딥러닝이 가장 좋은 성과를 보이는 분야 중 하나가 컴퓨터비전이라면, 다른 하나의 분야는 자연어처리(natural language processing, NLP)이다. 대화형 로봇에서부터, 음성인식 스피커, 번역 서비스가 모두 딥러닝을 통해 자연어처리가 가능해지면서 높은 성과를 보여주고 있다. 이번 강의에서는 자연어처리에 활용되고 있는 알고리즘인 RNN(recurrent neural network)와 LSTM(long-short term memory)등을 학습하게 된다. 음성을 텍스트로 변환하거나, 음악을 합성하는 시프로그래를 실습해볼 수 있을 것으로 보인다.

기간 | 미정

마치며

지난주, 저희 편집진 중 한 명이 한 유명 온라인 AI연구 모임의 글을 읽던 중 조금은 낯선 글 하나를 발견했습니다. 고등학생이 글을 하나 올렸습니다.

“안녕하세요, 딥러닝 공부 중인 고등학생입니다.”로 시작한 글에는 “학교 소논문 R&E(Research and Education)로 인터넷에서 4개의 차종(아반떼, 그랜저, 벤츠e class, 모닝)을 각각 1000장을 다운받고 수작업으로 600장을 선별한 후 500장을 학습시켰다”라는 내용이 게재됐습니다. 학습의 정확도는 52%, 그래서 이 학생은 자신이 세운 간단한 모델 구조와 적은 데이터 양을 만족치 않은 결과를 낸 원인으로 생각하며, 선배 연구자들에게 문의를 했습니다. 감사하게도 ‘고등학생 딥러닝 연구자’의 질문에는 자발적인 충고와 제언이 이어졌습니다. 따로 이야기해 보니, 대전의 고등학교 3학년 재학생(1998년생)이었습니다. 이름은 신승윤. 또래들과 마찬가지로 농구를 좋아하는 이 학생은 또 다른 취미로 프로그래밍을 갖고 있는 터라, 바쁜 수험 생활 중에도 딥러닝에 대한 관심의 끈을 놓지 않고 있었습니다.

승윤 군이 딥러닝에 관심을 갖게 된 것은 컴퓨터가 사진을 인식하고 설명문을 작성하는 모습을 봤을 때 부터 입니다. “이게 너무나 신기해서, 그 때 부터 AI의 개념들을 찾아 보기 시작했어요.” 인문계이기에 주변에 딥러닝을 이야기할 친구들은 없다고 합니다. 그래서 코세라(Coursera)의 앤드류 응(Andrew Ng) 교수 강의를 듣고, 독학을 합니다. 그러다 모르면, 온라인 게시판을 이용해 AI 연구 선배들에게 자문을 구하고 있습니다. “겨우 탐구하는 수준”이라고 말하지만, 실제로 신 군은 간단한 딥러닝 모델들(CNN, RNN)을 구현해 보고, 새로운 데이터를 학습시킬 수 있다고 합니다. “논문만 봐서는 이해가 잘 안되는 것이 많고, 수학적 지식이 부족해 원하는 모델 구현은 버겁다”는 신 군은 대입을 위한 수시 접수 뒤, 칸 아카데미(Khan Academy)에서 선형대수학을 배울 생각을 하고 있습니다. 대학에서 딥러닝 뿐만 아니라, 컴퓨터 그래픽스, 운영체제, 네트워크 등 컴퓨터 분야의 광범위한 분야를 섭렵한 뒤, 스타트업으로 세상에 도전에 해 보고 싶다는 것이 승윤 군의 포부입니다.

승윤 군과의 이야기 하면서, “문제다, 사람이 없다”는 AI 업계의 푸념이 향후 몇 년안에는 사라질 수도 있겠다는 기대를 갖게 됐습니다. AI와 관련한 모두의 열린 채널이 되어, 저변을 넓히는 데 작은 기여라도 하고자 하는 카카오 AI 리포트 편집진은 다시 신발끈을 고정시키고 가을을 맞이합니다. 다음 호에 뵙겠습니다.

