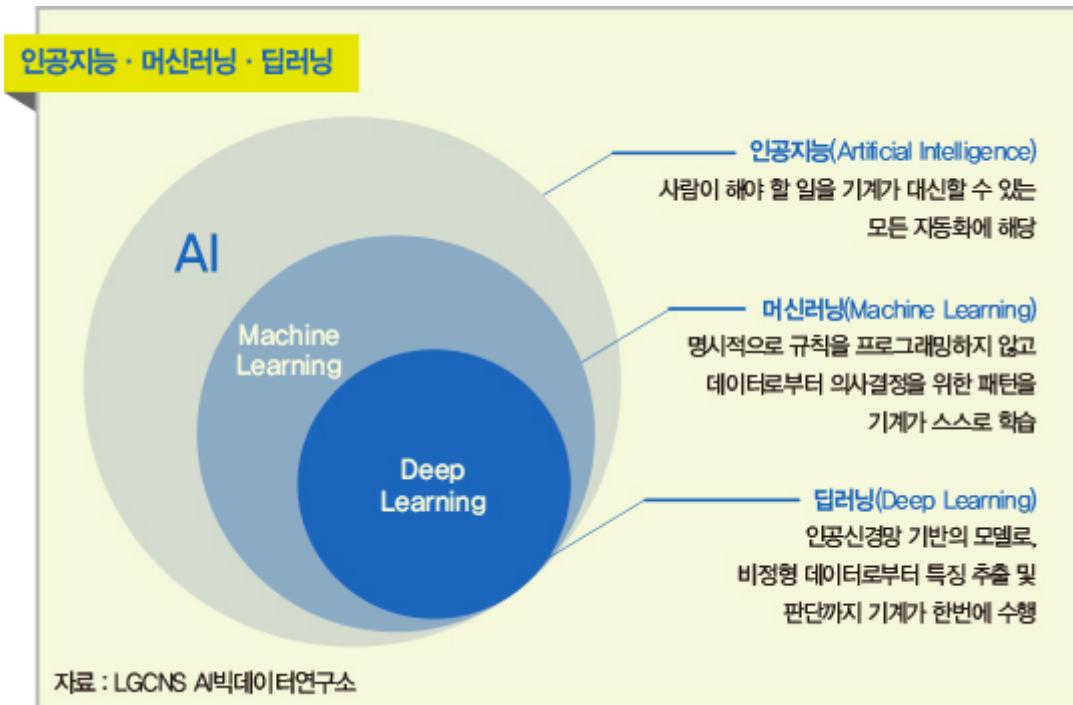


[AI·tech=LG CNS의 AI 이야기]

- 특징 분류하는 기존 방식은 예외 많아 신뢰성 낮아...신경망 기반 딥 러닝으로 돌파구 열려



[이주열 LG CNS AI빅데이터연구소장, 김명지 AI빅데이터연구소 책임] 최근 산업계에서 가장 화두가 되는 단어 중 하나는 바로 인공지능(AI)이다. AI는 넓은 의미에서 인간의 합리적인 사고나 행동을 모방해 자동화한 프로세스를 일컫는다. 간단히 생각하면 숫자와 연산 기호 버튼만 눌러 자동으로 값을 계산해 주는 계산기도 일종의 AI라고 볼 수 있다. 하지만 최근에 학계와 산업계에 불고 있는 AI 열풍은 이렇게 단순히 '규칙의 자동화'를 지칭하는 넓은 의미의 AI는 아닐 것이다.

요즈음 회자되는 AI는 좁은 의미로 딥 러닝 기반의 AI를 일컫는 경우가 대부분이다. 딥 러닝은 인공 신경망으로 이뤄진 모델을 활용해 기계가 의사 결정에 필요한 특징을 데이터로부터 알아서 추출하고 최종 판단을 내리는 기계학습(machine learning)의 한 방식이다. 하지만 회귀 분석이나 의사 결정 나무 등의 전통적인 머신 러닝 기법과는 조금 차이가 있다.

기계가 자동으로 강아지와 고양이를 구별하게 하려면 어떻게 해야 할까. 기계에 강아지와 고양이에 대한 우리의 지식을 전수할 방법을 생각해 보자. 고양이는 대체로 귀가 작고 뾰족하고 주둥이 길이가 상대적으로 강아지보다 짧은 특징이 있다. 또 박스를 던져 주면 고양이는 그 안에 들어가고 강아지는 박스를 물어뜯고 좋아한다. 이러한 정보를 활용해 볼 수 있지 않을까.



딥 러닝 이전의 AI

우리의 지식을 토대로 분류 규칙을 만들어 프로그래밍하면 자동화 시스템을 만들 수 있을 것이다. 사람은 강아지와 고양이를 구분하는 주요 특징들을 정리해 기계에 알려줘야 한다. 그런데 이대로 두면 몇 가지 문제가 발생한다.

예를 들어 강아지처럼 귀가 접힌 스코티시폴드, 박스 안에서 눈만 보이는 시베리안 허스키 사진이 있다고 하자.

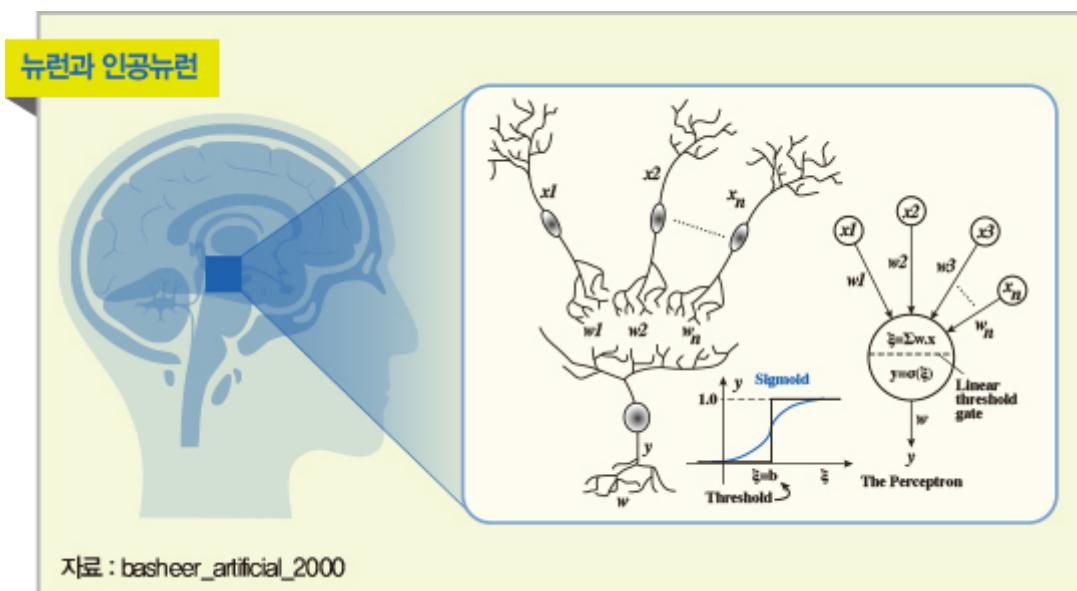
이 데이터들은 우리가 정의했던 강아지와 고양이의 특징에 부합하지 않는다. 이러한 특정 케이스에 대해 '예외 규칙을 추가하면 되지 않을까'라고 쉽게 생각해 볼 수 있지만 예외란 것이 언제 어떤 것이 발생할지 알 수 없고 그때마다 매번 규칙을 수정하거나 추가해 주는 공수가 많이 들어가게 된다. 편하기 위해 AI를 개발하는 것인데 더 귀찮은 상황이 발생할 수도 있는 것이다. 사람은 척 보면 이 친구들이 강아지인지 고양이인지 알 수 있는데 사람은 도대체 어떻게 이런 예외들에 대해서도 유연하게 대처할 수 있을까.

사람은 세상에 태어나 '고양이는 주둥이가 짧고 귀가 뾰족하고 박스를 좋아한다'와 같은 규칙, 사전적인 정의를 통해 고양이를 배우지 않았다. 동물 그림 카드를 보면서 '아 강아지는 이렇게 생겼구나' 또는 TV나 동화책에 나온 여러 동

물들, 지나가면서 마주친 고양이를 부모님이 '저기 귀여운 야옹이가 있구나'라고 말해 주는 것을 들으면서 배우게 된다.

우리는 사전적 정의나 규칙이 아니라 다양한 여러 사례들을 보고 귀납적으로 강아지와 고양이에 대한 정보를 습득한다. 말로는 구체화하기 어렵지만 대충 강아지와 고양이는 각각 이러한 특징들이 있다는 것을 깨닫는다.

그렇다면 기계에도 수많은 사례를 통해 구별하게 하면 다양한 예외에도 잘 대처할 수 있지 않을까. 딥 러닝은 바로 이러한 모티브를 기계 학습에 녹여낸 방법이다.



딥 러닝 기반의 AI

사람의 사고방식을 기계에 적용하기 위해 컴퓨터공학과 신경과학이 손을 잡는다. 사람에게는 '뉴런'이라는 신경 세포가 있어 세포의 한쪽에서 받아들인 전기 자극 정보를 다른 쪽 끝에서 다음 세포로 전달한다. 이 모양을 그대로 본떠 만든 것이 '인공 뉴런'으로, 입력 노드로부터 받아들인 데이터를 가중합 연산을 한 뒤 비선형 함수를 적용, 정보를 가공해 다음 인공 뉴런으로 넘기는 역할을 한다.

이런 인공 뉴런을 다양한 방식으로 여러 층 쌓아 연결하게 되면 딥 러닝의 기본 구조인 '인공 신경망'이 된다. 그리고 수많은 강아지와 고양이 사진을 인공 신경망에 보여주면 자동으로 강아지와 고양이 분류를 위한 최적의 연산 모델을 찾아낸다. 여기엔 귀가 어떻다든지, 주둥이 길이가 어떻다든지 하는 인간의 지식은 필요 없다. 딥 러닝 모델은 데이터를 통해 자동으로 필요한 특징을 찾아내고 분류를 수행한다.

기계 자동화를 위한 방법은 점점 사람의 개입이 줄어드는 방향으로 발전하고 있다. 예전엔 어떻게 해서든지 인간이 알고 있는 지식(규칙)을 기계에 전수하려고 했다면 이제는 인간이 사고하는 방식 그 자체를 기계에 알려주고 데이터를 제공하게 됐다. 최근엔 딥 러닝 기반의 AI가 다양한 영역에서 인간의 성능을 뛰어넘었다는 기사도 많이 볼 수 있다. 기계는 과연 어디까지 발전할 수 있을까.

[본 기사는 한경비즈니스 제 1293호(2020.09.07 ~ 2020.09.13) 기사입니다.]

[AI 이야기]

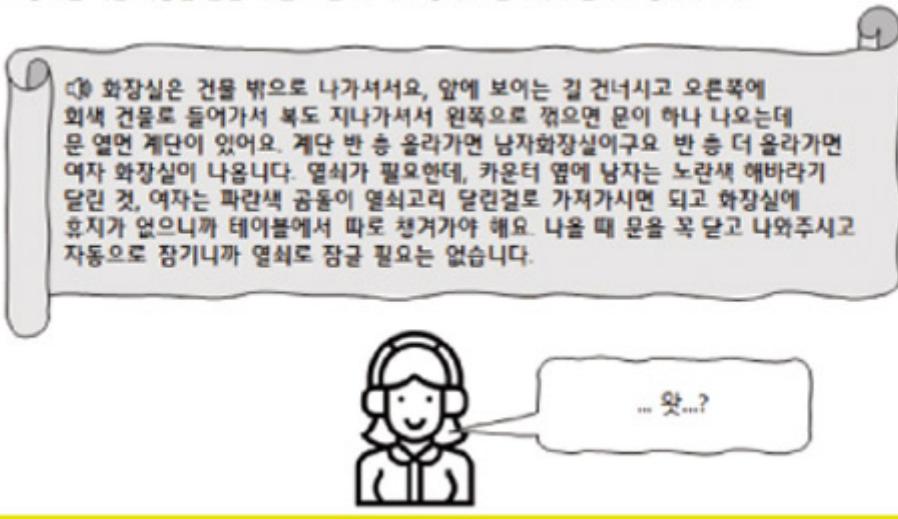
입력 데이터의 어텐션 스코어 계산해 활용...'설명 가능한 인공지능' 구축에도 도움

〈그림1〉 짧은 문장은 한번 들는 것만으로 쉽게 번역할 수 있다



(그림: 짧은 문장은 한 번 들는 것 만으로 쉽게 번역할 수 있다)

그렇다면 다음 내용을 한번 다 듣고 난 뒤 바로 영어로 번역해야 한다고 생각해보자.



[한경비즈니스 칼럼=이주열 LG CNS AI빅데이터연구소장, 김명지 AI빅데이터 연구소 책임] 사람은 최근의 내용은 뚜렷하게 기억하지만 시간이 지난 사건일수록 기억이 가물가물해진다. 어제 한 일에 대해서는 구체적으로 떠올릴 수 있지만 1~2년 전의 일과가 어땠을지 바로 기억할 수 있는 사람은 거의 없을 것이다. 그때의 일기·메일·기록한 일정 등을 다시 찾아보며 어떤 일이 있었는지 되짚어 봐야 한다.

인공지능(AI)도 마찬가지다. 순환 신경망(RNN)은 시간 순서에 따른 데이터를 처리하는 인공 신경망으로, 긴 시간에 대해 누적된 데이터가 입력되면 먼 과거의 내용을 잘 반영하기 어렵다는 점이 한계다. 예를 들어 기계 번역 과제를 생각해 보자. 간단한 문장은 한 번 들는 것만으로 쉽게 번역할 수 있다.

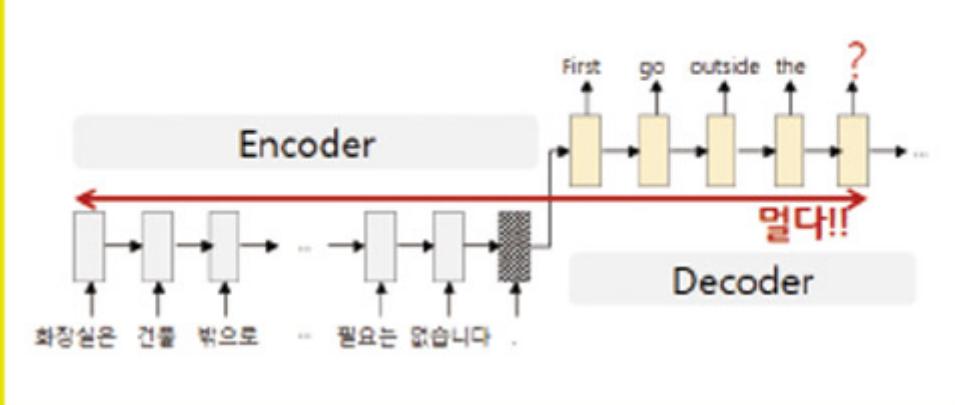
그러면 [그림1] 내용을 한번 다 듣고 난 뒤 바로 영어로 번역해야 한다고 생각해 보자.

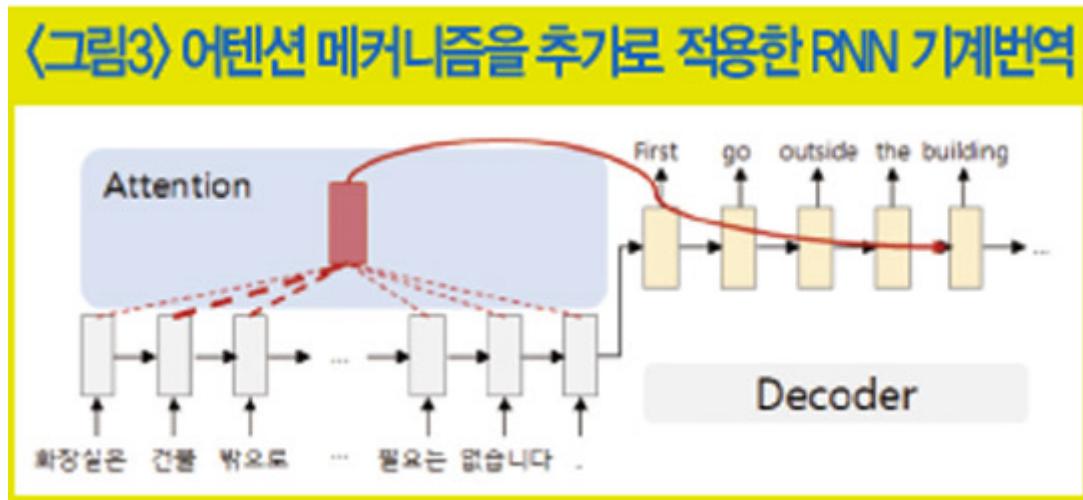
다시 들을 기회가 없다고 가정할 때 한번 듣는 것만으로 전부 번역할 수 있을까. 내용 자체는 어렵지 않다. 하지만 다량의 내용이 담긴 긴 문장(또는 문단)을 딱 한 번만 들고서는 번역을 정확하게 할 수 없을 것이다. 우리가 긴 문장을 번역할 때 한 단어 또는 구절을 영어로 옮길 때마다 필요한 내용을 한국어 원문에서 되짚어 가며 옮기곤 한다.

한국어와 영어의 어순이 다르니 뒷부분을 봤다가 다시 앞부분을 참고하기도 하고 여러 어절을 한 영어 단어로 번역하거나 반대로 한 어절을 여러 구로 번역하기도 할 것이다. 문장이 문단이 되고 문서가 될수록 더욱 더 여러 번 원문을 재참조해야 한다.

인간의 기억력에는 한계가 있어 모든 입력 내용을 한 번 읽고 결과를 한 번에 생성하는 것보다 그때그때 재확인하면서 필요한 부분을 보는 것이 더 도움이 된다. 사람이 원문(input data)을 전체적으로 훑으면 중요한 부분을 다시 참고 하듯이 인공 신경망 학습에도 이러한 모티브를 녹여낼 수 없을까.

〈그림2〉 기본 RNN의 인코더 · 디코더를 활용한 기계번역





어텐션 메커니즘(attention mechanism)

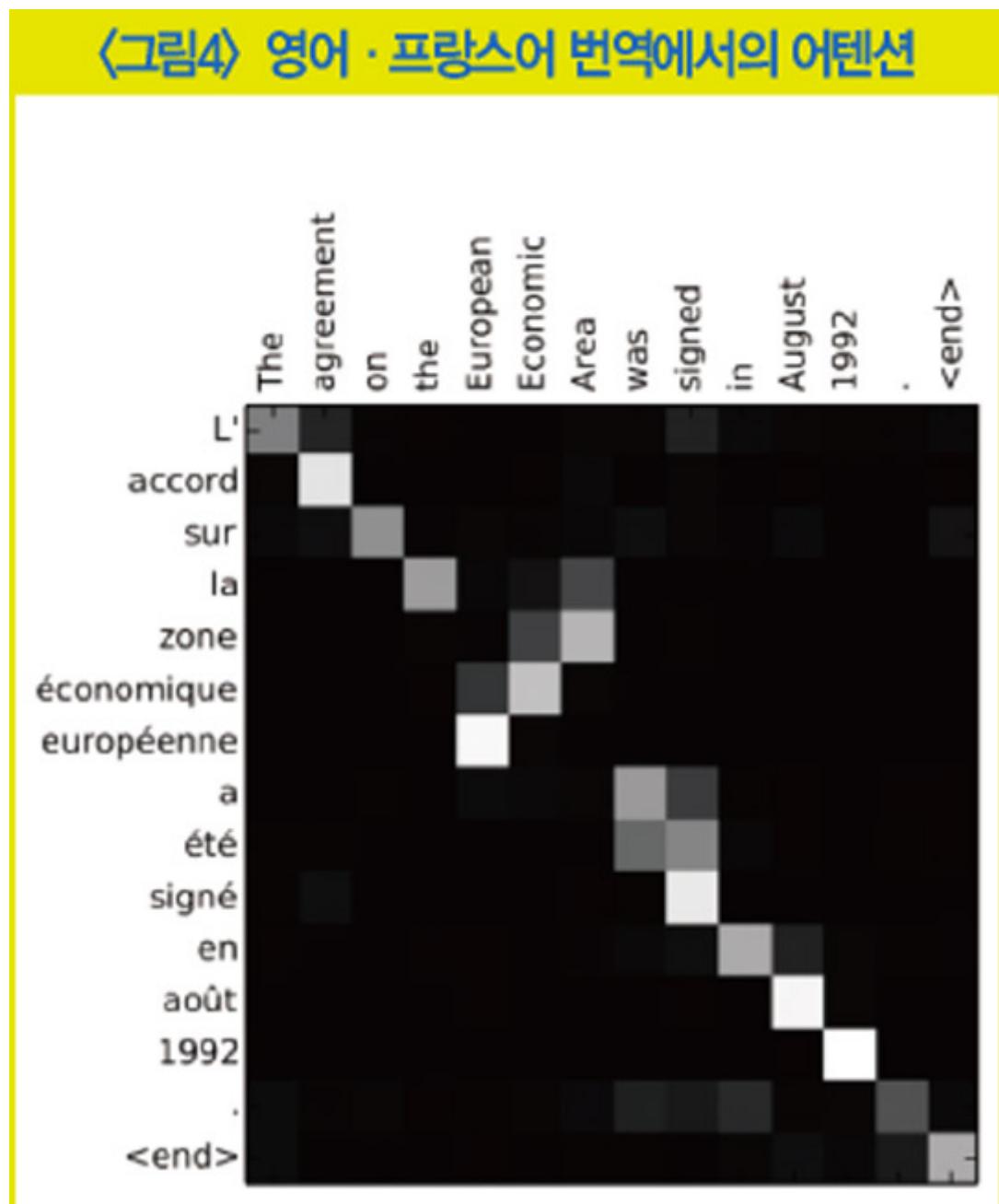
학교에서의 수업 시간, 모든 과목의 모든 내용을 빠짐없이 다 주의 깊게 들었던 학생은 많지 않을 것이다. 딴짓을 하다가 선생님 질문에 맥락을 이해하지 못하고 벼벽거리면 "집중 좀 합시다"란 꾸중을 들을 것이다.

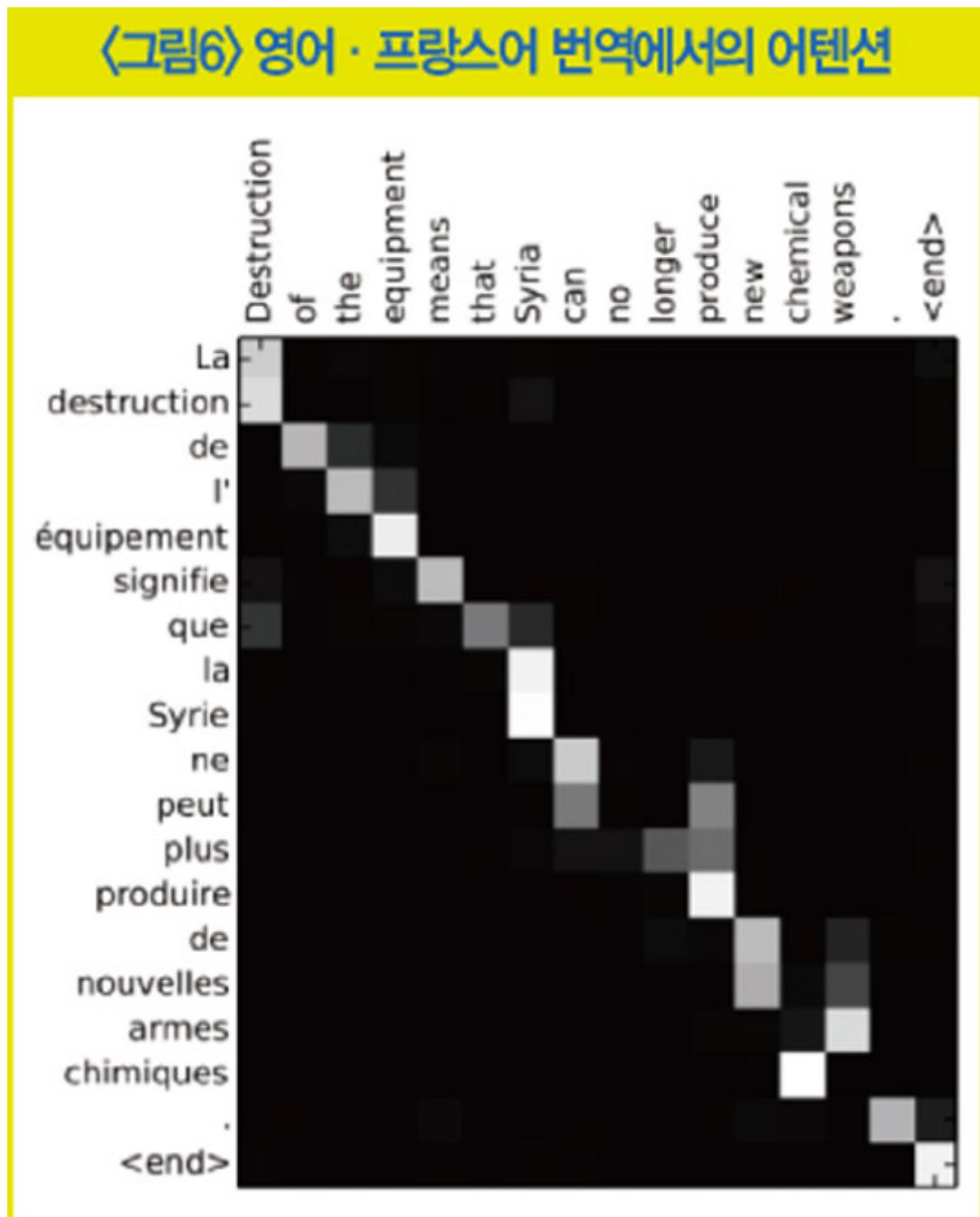
인공 신경망 모델에서의 집중(attention)이란 무엇일까. 인공 신경망이 수행하는 '집중', 어텐션 메커니즘에 대해 알아보자. 어텐션 메커니즘은 인공 신경망이 입력 데이터의 전체 또는 일부를 되짚어 살펴보면서 어떤 부분이 의사 결정에 중요한지 중요한 부분에 집중하는 방식이다. 우선 어텐션 없이 기본 RNN만으로 구성한 기계 번역 모델이 있다면 [그림2]와 같을 것이다.

RNN은 입력 문장의 단어 하나하나를 누적 압축해 인코딩하고 있다가 모든 문장이 다 들어오면 영어로 한 단어씩 번역을 수행(디코딩)한다. 이때 디코더가 참고하는 문맥은 [그림2]의 빗금 영역에 해당하는, 입력문이 전부 압축된 하나의 벡터다. 이 벡터는 긴 문장을 모두 누적하고 있지만 문장 앞부분의 내용은 너무 압축된 나머지 정보를 거의 잊어버린 것이나 마찬가지다. 여기에 어텐션 메커니즘을 엿어 번역 시 원문을 다시 재참조하며 현재 디코딩할 단어와 연관된 중요 부분에 집중하게 하면 어떨까. [그림3]과 같이 구성해 볼 수 있다.'빌딩(building)'이란 단어를 생성할 때 인공 신경망은 전체 한국어 입력 문장을 되짚어 보며 현재 단어를 디코딩하기 위해 중요한 부분이 어디일지 생각하게 된다. 그 결과 수많은 입력 단어 중 '건물'에 해당하는 단어에 조금 더 주의를 기울일 필요가 있다고 판단, 해당 단어에 조금 더 집중해 전체 입력을 다시 한번 재조정한 입력 데이터 인코딩 벡터를 만든다. 이렇게 하면 입력 문장이 매우 길어진다고 해도 전체 문맥을 골고루 참고할 수 있게 되므로 더 좋은 번역을 할 수 있다.

어텐션 스코어(attention score)

이때 '중요한 단어에 집중한다'는 것은 어텐션 스코어를 계산한다는 것인데 어텐션 스코어는 인공 신경망 모델이 각 인코딩 타임스텝마다 계산된 특징을 가지고 자동으로 계산하는 0~1 사이의 값이다. 어떤 부분은 더 집중해 봐야 하고 (1에 가까운 스코어), 어떤 부분은 지금은 중요하지 않으므로 대충대충 살피도록(0에 가까운 스코어) 하는 것이다. [그림3]에는 붉은 점선의 굵기로 어텐션 스코어가 표시돼 있다. 어텐션 스코어는 각 단어에 대한 주의 집중 가중치라고 볼 수도 있다.





콘텍스트 벡터(context vector)

어떤 부분을 더 살펴보고 어떤 부분은 대충 봐도 좋을지에 대해 어텐션 스코어를 구하고 나면 현재 디코딩할 단어와의 관련성을 반영해 다시 입력 문장을 인코딩하게 되는데, 이는 중요도에 따라 전체 문맥의 정보를 잘 반영하고 있다고 해서 콘텍스트 벡터라고 부른다. 어텐션 스코어와 콘텍스트 벡터를 만드는 방식은 여러 가지가 있다. 하지만 중요한 것은 스코어 계산에 필요한 수식이 아니라 어텐션 메커니즘이 매번 디코딩마다 직전 단계의 벡터뿐만 아니라 과거의 모든 데이터의 특징을 고려한다는 점이다.

또 하나의 포인트는 딥러닝 모델이 '스스로' 집중할 영역을 파악한다는 것인데, 딥러닝 모델은 데이터를 더 잘 맞추도록 학습하는 과정에서 어떤 부분이 중요 한지 사람이 알려주지 않아도 알아서 집중할 영역을 찾아낼 수 있다. 딥러닝을

활용한 기계 번역을 위해 어텐션 메커니즘을 처음 도입한 논문에 보면 이런 자료가 있다.

[그림4]는 영어 문장을 프랑스어로 번역하는 과제를 할 때 각 단어 번역 시 인공 신경망이 어떤 영어 단어 쪽에 집중했는지 어텐션 스코어를 시각화한 그림이다. 하얀색으로 표시될수록 딥러닝 모델이 해당 단어를 더 주의 깊게 봤다는 뜻이 되고, 이는 사람이 알려준 것이 아닌 신경망 스스로가 학습한 집중 패턴이다. 그림상으로도 알 수 있듯이 영어와 프랑스어는 대강 비슷한 어순을 가지고 있다.

특이한 점은 영어의 'European Economic Area'가 프랑스어의 "zone économique europeene"에 해당하는 것을 들 수 있는데, 이 부분은 영어와 프랑스어 어순이 반대다. 딥러닝 번역 모델의 어텐션 스코어를 보면 신기하게도 이 부분에서 순서를 반전시켜 가며 주의를 기울이고 있다. 완전히 동일한 어순을 가지며 단어 간 일대일 매칭이 되는 언어끼리의 번역이 아니라면 그때그때 유연하게 집중해야만 하는데 이때 어텐션 메커니즘이 그 역할을 훌륭히 해냈다. 예시를 하나 더 살펴보자.

[그림6]에서 프랑스어 'La destruction'란 단어를 번역할 땐 영어의 'Destruction'에 집중하고 'la Syrie'를 번역할 땐 영어의 'Syria'에 집중한 것을 볼 수 있다. 프랑스어가 단어 앞에 관사를 붙이는 점이 영어와 다른 특징이라고 볼 때 딥러닝 모델은 해당 언어별 특징을 잘 이해하고 집중할 부분을 선택한 것을 확인할 수 있다.

〈표1〉 예시

AI : "아무개씨, 당신은 유죄입니다! 감옥에서 종신형을 선고합니다."
나 : "...? 왜... 왜죠? 저는 납득할수가 없어요! 제가 유죄라니요!"
AI : "이유는 설명할 수 없습니다. 하지만 제 판단은 정확합니다!"

XAI로서의 어텐션

예제에서 알 수 있듯이 어텐션 메커니즘은 기계가 판단할 때 중요하게 생각하는 부분을 사람에게 알려주는 역할도 한다. 이는 설명 가능한 인공지능(XAI : eXplainable AI)으로서의 기능이라고 볼 수 있다. 딥러닝 기반의 AI는 일반 머신러닝 기반이나 전통적 런지 기반의 프로그래밍에 비해 예측 정확도는 높지만 그 모델이 너무 복잡하고 해석하기 어렵다는 단점이 있다. 이에 따라 AI의 추론 결과를 해석하는 XAI는 오늘날 매우 중요한 영역이다. 딥러닝 기반 AI의 이러한 단점은 특히 법률·의학·금융 등 민감한 내용을 다루는 도메인에서 문제가 되곤 한다. 인공 신경망의 무수한 파라미터를 사람이 이해할 수 있는 방법으로

표현하기가 쉽지 않기 때문이다.

예를 들어 미래에 AI 판사가 등장해 인간 판사를 대체할 수 있을 수준이 됐다고 가정해 보자. AI 판사는 아주 실력이 좋고 유능하면서도 24시간 일할 수 있고 공정하며 뇌물 수수의 유혹에도 흔들리지 않기 때문에 미래의 인간은 전적으로 AI의 판단에 결정을 맡기게 됐다. 하지만 다짜고짜 AI 판사가 여러분에게 [표1]과 같이 선언했다고 하자. 당연히 납득할 수 없을 것이다.

AI 판사가 평범한 인공 신경망으로 이뤄져 있다면 죄명이 무엇인지, 무엇을 얼마나 잘못했기에 이런 판단을 내렸는지 설명할 수 없다. 반면 인간 판사는 이러한 점을 설명해 줄 수 있다. 왜 이렇게 판단했는지, 죄명은 무엇이고 어떤 점 때문에 형벌의 강도를 심하게 또는 약하게 정했는지 등을 설명해 줄 수 있다. 벌을 달게 받는다는 게 쉬운 일은 아니지만 그래도 설명이 있다면 납득하고 받아들일 수는 있을 것이다.

어텐션은 인공 신경망의 이러한 설명 부족 문제를 일부 해소해 줄 수 있다. 물론 사람이 알아들을 수 있는 말로 '이렇고 저렇기 때문에 그렇게 판단했다'고 알려 주는 것은 아니지만 해당 결정을 내릴 때 '어떤 부분에 집중해 판단했는지'를 시각화해 보여 줄 수 있다.

〈그림7〉 소비자 만원 주제 자동 분류 예 (LGCNS, 2017)

텍스트에서의 어텐션

▶ input	alpha
신용/NNG	0.179
카드/NNG	0.198
할/XSV+ETM	0.034
부/XPN	0.047
결제/NNG	0.251
취소/NNG	0.146
관련/NNG	0.081

- ▶ 모형정답 : (1) 금융
- ▶ 모형정답 : (2) 정보통신서비스
- ▶ 모형정답 : (3) 의류·섬유신변용품

▶ input	alpha
스마트폰/NNP	0.043
하위/NNG	0.063
계약/NNG	0.120
개통/NNG	0.196
월회/NNG	0.196
거부/NNG	0.372

- ▶ 모형정답 : (1) 정보통신서비스
- ▶ 모형정답 : (2) 정보통신기기
- ▶ 모형정답 : (3) 도서·음반

〈그림8〉 이미지 어텐션



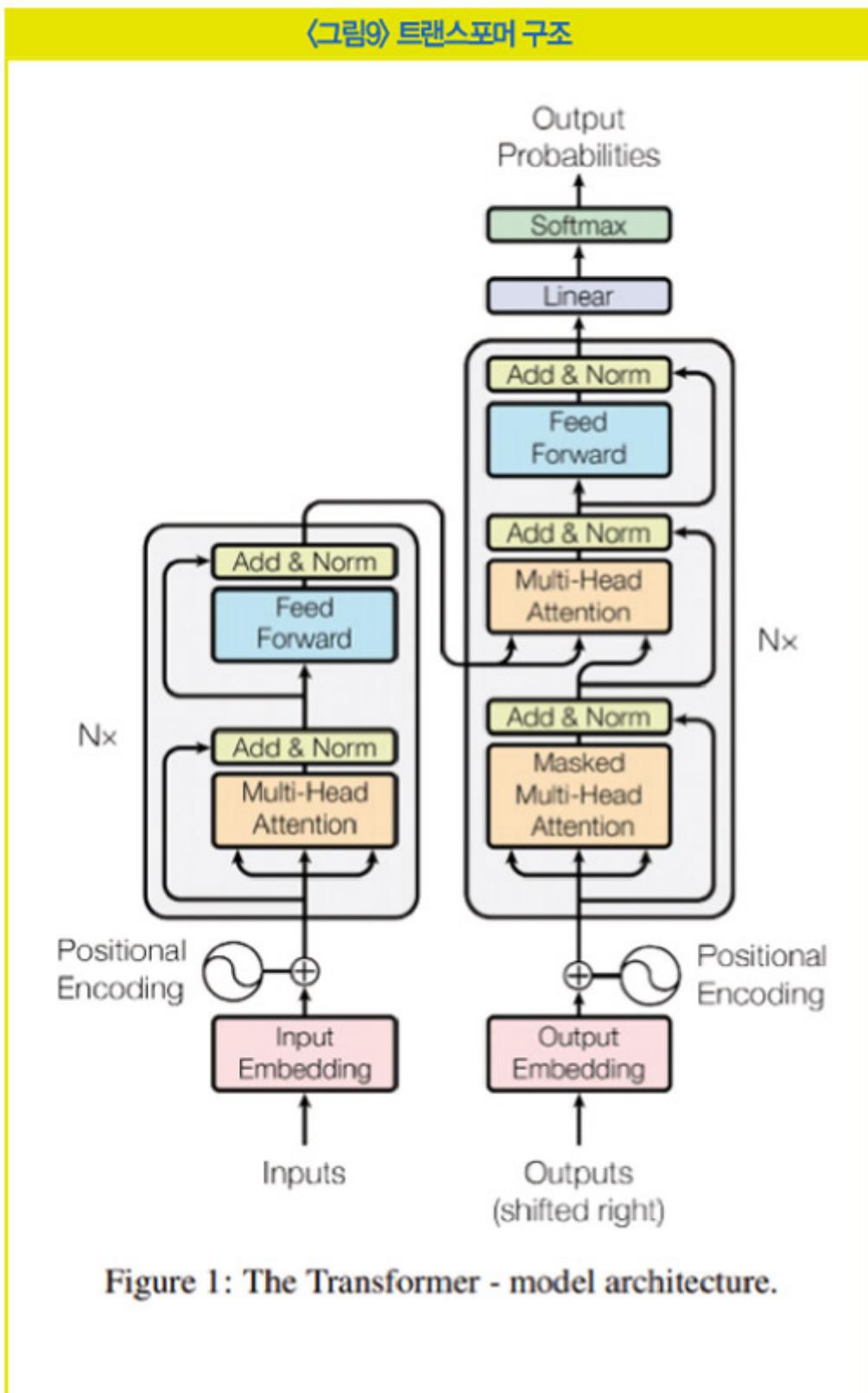


Figure 1: The Transformer - model architecture.

[본 기사는 한경비즈니스 제 1312호(2021.01.18 ~ 2021.01.24) 기사입니다.]

1. [홈](#)
2. [한경BUSINESS](#)

AI 사이언티스트의 숙명 '튜닝'... 학습 모델 최적화에 수작업 필수

[페이스북 공유하기](#) [트위터 공유하기](#) [카카오톡 공유하기](#) [공유옵션 더보기](#)

공유하기

[페이스북](#) [트위터](#) [카카오톡](#) [네이버](#) [밴드](#) [다음카페](#)

<https://magazine.hankyung.com/business/article/202101061704b> URL 복사

네이버 채널 구독

[공유하기 레이어 닫기](#)

[폰트크기조정](#)

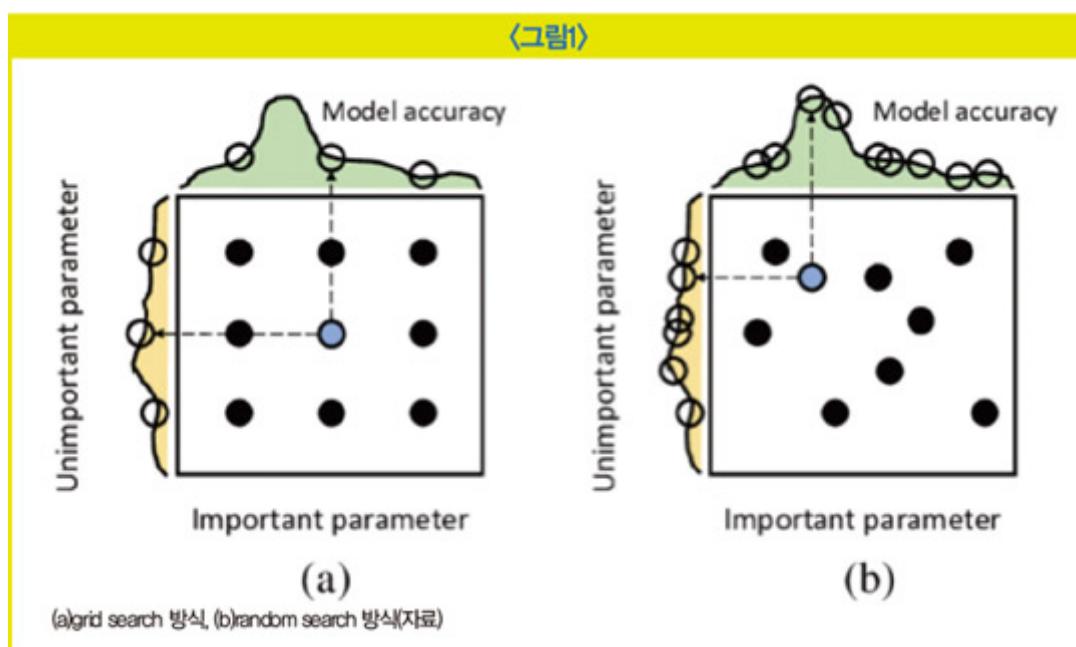
[폰트크기 가장 작게](#) [폰트크기 작게](#) [폰트 기본크기](#) [폰트크기 크게](#)

[폰트크기 가장 크게](#)

입력2021.01.06 09:26 수정2021.01.06 09:26

[AI 이야기]

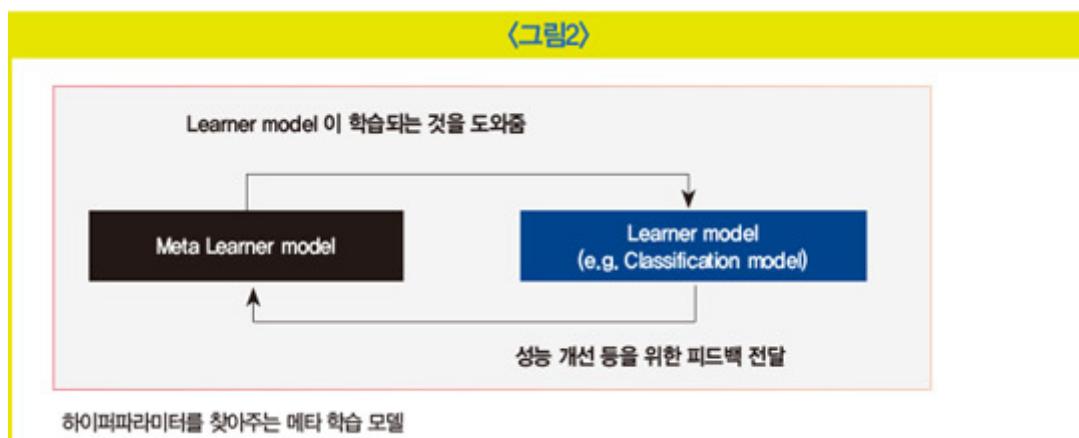
기계 학습마저 자동화한 '오토ML' 주목... 스스로 반복 실험 통해 최적 파라미터 등 찾아



[한경비즈니스 칼럼=이주열 LG CNS AI빅데이터연구소장, 김명지 AI빅데이터 연구소 책임] 인공지능(AI) 사이언티스트의 길을 걷다 보면 누구나 튜닝(tuning)의 장벽과 마주하게 된다. 튜닝은 현재 실험의 결과 양상을 보고 문제점을 진단하고 AI 모델을 조금 더 나은 방향으로 만들기 위해 실험을 개선하는 것을 말한다. 여기에는 아키텍처를 변경하거나 하이퍼파라미터를 조절하거나 하는 등의 역할이 포함된다. 하지만 어떤 현상이 나타났을 때 어떤 조치를 취해야 효과적인지에 대한 매뉴얼이 딱 정해져 있는 것은 아니고 있다고 하더라도 현장의 다양한 태스크에 꼭 들어맞지는 않는다. 왜 이런 결과가 나왔는지 해석하는 것도 사람의 뜻이기 때문에 우리는 문제의 원인을 유추하고 이에 기대 현상을 개선하기 위한 방법을 찾아야 한다. 따라서 실험을 진행하는 사람의 탄탄한 이론 배경과 함께 경험과 노하우까지 풍부해야만 불필요한 실험의 반복 횟수를 줄일 수 있다.

튜닝해야 할 대상은 너무나 다양하다. 딥 러닝 모델로 한정짓는다고 하더라도 FNN·CNN·RNN·트랜스포머 등등 어떤 계열의 모델 구조를 이용하는 게 좋을까. 인공 신경망의 층수는? 한 층에 들어갈 인공 뉴런의 수, 콘볼루션 필터 사이즈와 필터 수는? 얼마나 성큼성큼 학습시키는 것이 좋을까(learning rate)? 한 번에 학습할 데이터의 수는 어느 정도가 적당할까(mini-batch size)? 몇 번이나 반복해서 보여줘야 적당할까(epoch)? 어떤 최적화 기법을 쓰고(optimizer), 손실 함수는 어떤 것을 쓰는 게 효과적이며(cost function) 활성화 함수는 무엇이 좋을까?

모델 학습을 위해 사람이 결정해야 하는 설정이 한두 가지가 아니다. 딥 러닝 기반의 AI가 머신러닝이나 룰 기반의 AI보다 수작업의 공수가 적다고 했지만 여전히 사람의 개입이 필요한 부분이 있다. 스스로 척하면 척, 알아서 학습할 수 있는 AI는 없을까.



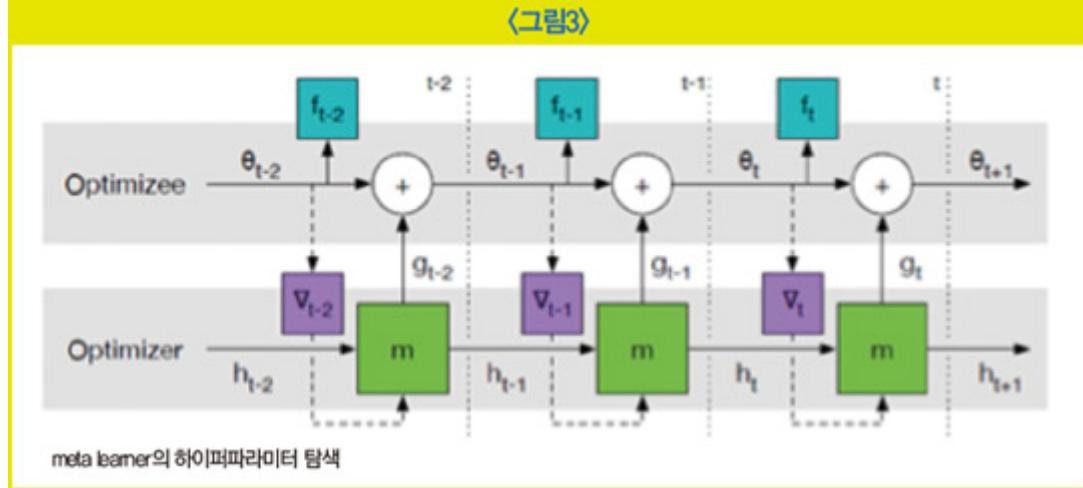
스스로 진화하는 인공지능, 오토(Auto)ML

AdChoices
광고

사람이 수많은 시행착오를 겪으며 스트레스를 받지 않아도 자동으로 적절한 AI가 학습되도록 도와주는 기법이 있다. 이를 오토(Auto)ML(Automated Machine Learning), 말 그대로 자동화된 기계 학습이라고 부른다. AI가 스스로 진화한다는 표현이 좀 과장되기는 했지만 여기서 말하는 진화란 한정적인 의미다. 여기서는 '특정 태스크를 위한 모델 학습'에 한해 사람이 주기적으로 실험에 개입하지 않아도 AI 스스로가 반복 실험을 통해 성능을 개선하는 것을 말한다. 가만히 두면 AI 스스로가 똑똑해져 이것도 배우고 저것도 배우며 결국엔 사람을 지배하는 스토리의 진화는 결코 아니다.

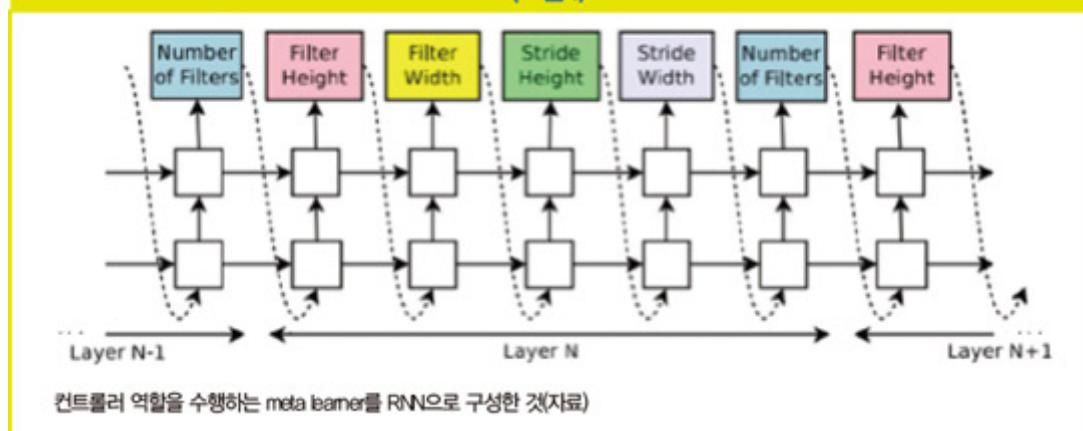
오토ML의 역할은 크게 세 가지로 나눠 볼 수 있다. 첫째는 AI 모델을 학습하기 위해 데이터에서 중요한 특징(feature)을 선택하고 인코딩하는 방식에 대한 특징 엔지니어링(feature engineering) 자동화다. 둘째는 AI 모델 학습에 필요한 사람의 설정들, 하이퍼파라미터를 자동으로 탐색해 주는 것이다. 셋째는 AI 모델의 구조 자체를 더 효율적인 방향으로 찾아주는 아키텍처 탐색이다. 이미 딥러닝 모델은 비정형 데이터를 깊은 인공 신경망에 태워 자동으로 특징을 추출한다는 장점이 있기 때문에 그중 특징 엔지니어링에 대해서는 별도로 다루지 않도록 한다.

〈그림3〉



meta learner의 하이퍼파라미터 탐색

〈그림4〉



컨트롤러 역할을 수행하는 meta learner를 RNN으로 구성한 것(자료)

하이퍼파라미터 탐색 자동화

딥 러닝 모델 학습에 필요한 하이퍼파라미터는 다양한 종류가 있다. 모델의 파라미터 업데이트를 얼마만큼 큰 단위로 할지 결정하는 학습률(learning rate), 데이터를 얼마나 쪼개 학습할지의 단위인 미니배치 사이즈(mini-batch size), 데이터를 몇 번 반복 학습할지에 대한 단위 에폭(epoch), 이 밖에 모멘텀이라든지 콘볼루션 필터의 수, 스트라이드 등등 사람이 설정해 주지 않아도 자동으로 결정되는 값은 하나도 없다. 많은 경우 딥 러닝 학습 프레임워크 (TensorFlow, PyTorch 등)에서는 기본적으로 잘 작동하는 설정을 디폴트로 제공하고 있다. 하지만 기본 설정으로도 학습이 잘되지 않는다면 실험 결과를 살핀 뒤 하이퍼파라미터를 조금씩 튜닝해 줘야 한다. 이게 워낙에 반복적이고 공수가 많이 드는 작업이다 보니 기존에 이미 여러 가지 하이퍼파라미터의 조합을 찾고자 하는 시도가 있었다. 자주 쓰이는 것 두 가지만 들자면 그리드 서치(grid search)와 랜덤 서치(random search) 방식이 있다.

그리드 서치 방식은 최적화할 하이퍼파라미터의 값 구간을 일정 단위로 나눈 후 각 단위 조합을 테스트해 가장 높은 성능을 낸 하이퍼파라미터 조합을 선택하는 방식이다. 단순하지만 최적화 대상이 되는 하이퍼파라미터가 많다면 경우

의 수가 기하급수적으로 많아져 탐색에 오랜 시간이 걸릴 수 있다. 또한 불필요한 탐색에 시간을 허비하기도 한다. 예를 들어 (a)에서 맨 왼쪽 열의 조합들은 굳이 세 번을 다 학습해 볼 필요가 없지만 그리드 서치 방식을 이용하면 어쩔 수 없이 다 탐색을 수행해야 한다. 반면 오른쪽의 랜덤 서치 방식은 랜덤하게 하이퍼파라미터의 조합을 테스트하는 방식인데 그리드 서치에 비해 비교적 빠르게 최적의 조합을 찾아내곤 한다. 이 두 가지 방식은 어찌 보면 경우의 수를 찾는 단순 탐색법에 불과한데 단순히 'for문'을 이용해 구현할 수도 있다. 하지만 최근의 오토ML 방식에서는 하이퍼파라미터도 모델을 통해 탐색하곤 한다.

주어진 태스크를 수행하는 학습 모델(learner model)이 본래 우리가 알고 있던 AI 모델이라면 이 AI 모델이 좋은 성능을 달성하기 위한 최적의 하이퍼파라미터의 조합을 찾아주는 메타 학습 모델(meta learner)이 별도로 있다. 메타 학습 모델은 대부분 RNN과 강화 학습을 활용해 최적의 하이퍼파라미터를 탐색한다.

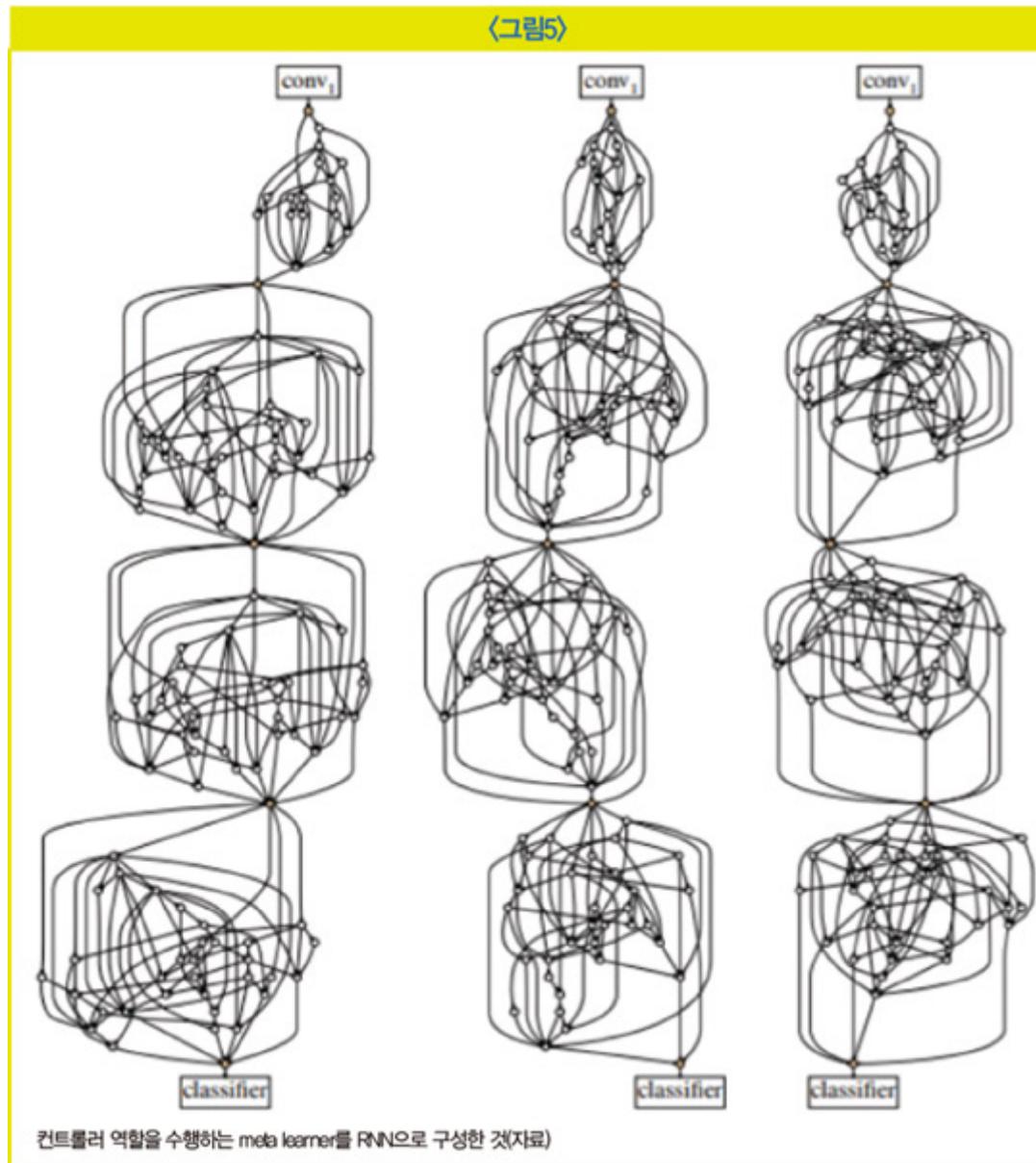
메타 학습 모델의 하이퍼파라미터 조합대로 학습한 학습 모델의 학습 성능 결과를 메타 학습 모델로 다시 전달하고 메타 학습 모델은 이를 또 개선하기 위한 다른 하이퍼파라미터 조합을 내며 학습 모델은 이 조합으로 또다시 학습한다. 이러한 과정을 반복하다 보면 최적의 조합을 찾아낼 수 있다. 이때 메타 학습 모델이 수행하는 학습에 대해 학습을 위한 학습이라는 뜻에서 '메타 학습' 또는 '런투런(learn to learn)'이라고 표현한다.

아키텍처 탐색 자동화

하이퍼파라미터뿐만 아니라 최적의 아키텍처를 찾아주는 방법도 있다. 아키텍처는 모델을 이루는 구조를 말하는데, 사람이 어떤 방식으로 모델 구조를 짤지 생각하지 않아도 자동 탐색을 통해 최적 구조를 찾을 수 있다. 특히 딥 러닝 모델은 인공 신경망을 활용하기 때문에 NAS(Neural Architecture search)라고 부른다. NAS도 마찬가지로 대부분 메타 학습 모델과 학습 모델로 이뤄져 있어 학습 모델이 본 과제를 수행하는 AI 모델이라면 메타 학습 모델이 어떤 구조의 신경망을 만들면 좋은지 아키텍처 구성을 고민한다.

메타 학습 모델은 역시 RNN과 강화 학습을 접목한 형식으로 구성해 볼 수 있다. 메타 학습 모델은 학습 모델의 인공 신경망 아키텍처가 어떻게 구성되면 좋을지 결정해 학습 모델의 태스크 수행 결과를 보상으로 활용한다.

이 밖에 진화 알고리즘이나 경사하강법을 기반으로 한 NAS 방식도 있다.



오토ML의 특징

오토ML을 활용하면 사람이 한 땀 한 땀 구조를 고민하고 하이퍼파라미터를 튜닝할 필요 없이 최적의 환경을 기계가 대신 결정해 줄 수 있다. 일반적으로 오토ML을 활용하면 사람이 고안한 모델 이상의 성능을 낼 수 있다. 기계는 사람이 생각도 못한 조합의 설정이나 구조를 시도해 볼 수 있고 기존의 설정 관습

이나 제약에 얹매이지 않기 때문이다.

는 2019년 AI 커뮤니티를 뜨겁게 달궜던 논문의 신경망 모델(S. Xie, et al., 2019, FAIR)인데, 오토ML을 통해 자동으로 탐색된 네트워크 구조다. 이 모델은 보기에는 임의로 엮인 실타래 같은 구조를 갖고 있지만 이미지 인식 과제에서 더 적은 FLOP 수로 사람이 만든 신경망 모델의 성능과 유사하거나 더 좋은 성능을 보였다. 기존의 NAS 방법들은 메타 학습 모델이 탐색할 아키텍처의 공간이 한정돼 있어 그중 최적의 아키텍처를 찾으려고 했다면 이 논문의 모델은 이러한 제약마저 없앤 진정한 의미의 NAS 방법을 고안했다. 이러한 모티브에서 랜덤 그래프 생성 방법론을 기반으로 아키텍처 탐색을 진행하자 처럼 기존 구성 방법의 틀을 완전히 깬 모델이 만들어졌다.

하지만 오토ML을 활용한다면 사람의 손길을 덜 필요로 하고 좋은 성능 결과를 얻을 수 있는 대신 기계가 다양한 시도를 해 보도록 오랜 시간을 기다려야 한다. 또한 고품질의 하드웨어 스펙이 뒷받침돼야만 이러한 창조적인 시도를 지원할 수 있다. 학습 모델과 메타 학습 모델이 동시에 발적으로 학습해야 하기 때문이다. 결론적으로 사람이 하이퍼파라미터를 찾거나 구조를 고민하기 귀찮다면 기계에 시간과 비용을 투자해야 한다.

마무리

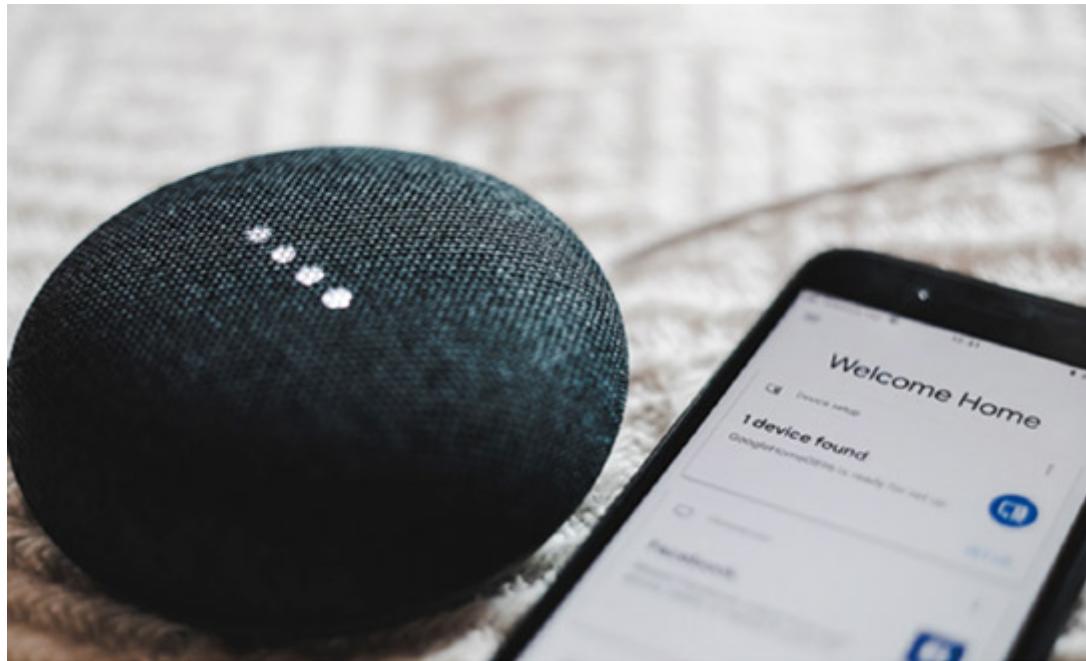
딥 러닝은 자동화의 패러다임을 바꿨다. 기존의 자동화 프로그래밍이 인간의 지식을 프로그래밍 언어로 명문화하여 기계에 주입하는 방식이었다면 딥 러닝은 인간의 사고방식 자체, 즉 인공적인 신경망을 기계에 구현한 뒤 수많은 데이터를 보여주며 말로 표현할 수 없는 지식을 학습시켰다. 오토ML은 여기에 다시 한 번 자동화를 추가하고 있다. 이제는 사고방식이 탄생될 수 있는 환경만 조성해 주면 기계가 알아서 나머지를 수행하게 된다. 앞으로의 AI 연구에서 사람의 역할은 태스크 수행을 위한 최적의 뇌 구조가 만들어질 수 있도록 기계를 지원하고 결과를 지켜보는 것이 되지 않을까.

[본 기사는 한경비즈니스 제 1310호(2021.01.04 ~ 2021.01.10) 기사입니다.]

© 매거진한경, 무단전재 및 재배포 금지

[AI 이야기]

-인공 신경망에 언어 데이터 넣어 ...문장을 토큰 단위로 자르고 벡터·매트릭스로 변화



[한경비즈니스 칼럼=이주열 LG CNS AI빅데이터연구소장, 김명지 AI빅데이터 연구소 책임] 옛날의 인류는 정보 교환을 위해 사람과 사람의 의사소통에 의존 할 수밖에 없었다. 문자가 생긴 뒤로는 정보를 기록할 수 있게 됐고 사람들은 궁금한 것을 책 등의 기록물을 보고 해결할 수 있었다. 요즈음 Z세대라고 불리는 초등학교 저학년 정도인 아이들은 컴퓨터보다 모바일 기기에 더 익숙하다. 이들은 질문이 생기면 인공지능(AI) 스피커에 음성으로 물어본다(짱구야~ ○○에 대해 알려줘~). AI 스피커에게 물어본다면 컴퓨터나 모바일 기기를 켜 자판 을 치고 검색할 필요도 없이 바로 대화로 답변을 얻을 수 있다. AI 스피커는 우리 가 무엇을 궁금해 하는지 어떻게 인식하고 답변해 줄까.

〈그림1〉

▶ “팀장님 이번 회식 소고기집 가도 돼요?”

▷ “이번 달 법카 얼마나 남았어?”

```

1 def is_ok(menu, card) :
2     if card.limit > menu.price :
3         return "OK"
4     else :
5         return "SORRY"
6
7 print is_ok(menu("소고기"), get_team_card(2018,5))
8

```

자연어(왼쪽)와 인공어(오른쪽)의 차이

자연어 이해(NLU)

자연어 이해 또는 자연어 처리(NLP : Natural Language Processing)는 말 그대로 자연어를 이해하거나 처리하는 기능에 대한 것인데, 그 주체는 ‘기계’다. 기계가 자연어를 이해한다는 것은 어떤 것일까. 용어를 하나씩 뜯어보기로 하자.

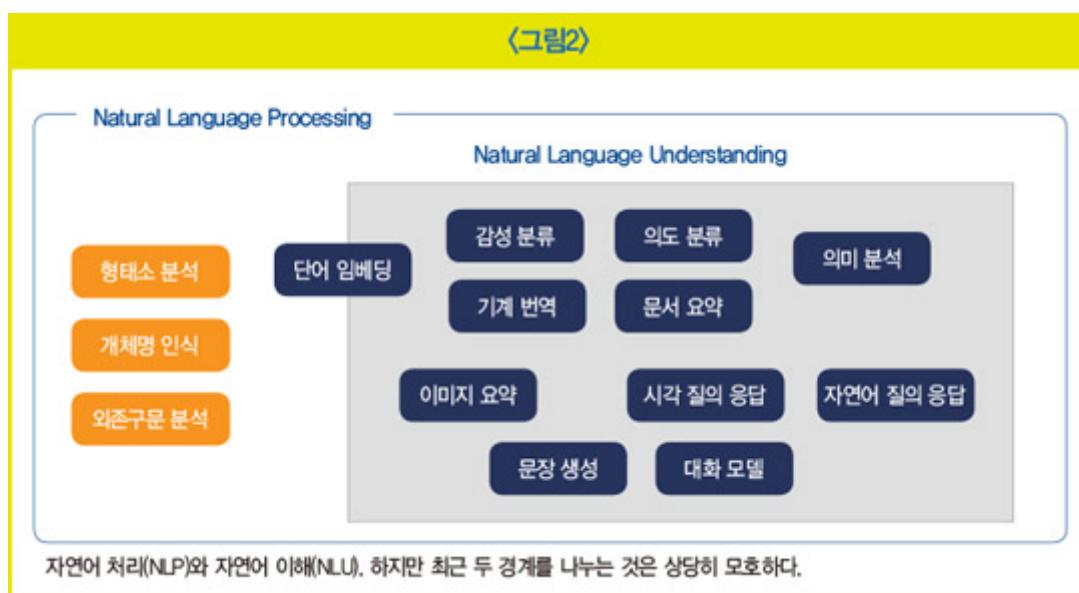
‘자연어(Natural Language)’는 사람들이 일상적으로 쓰는 언어로, 우리가 쓰는 한국어·영어·중국어·일본어 등이 있다. 자연어는 기원을 찾기 힘들며 자연 발생한다는 특징이 있다. 반대되는 개념으로 인공어(Constructed Language)가 있다. 인공어는 의도와 목적에 따라 인공적으로 만든 언어로, 프로그래밍 언어와 같은 기계어라든가 전 세계 공용 언어 사용을 위해 만들어진 에스페란토 등이 이에 해당한다.

‘언어(Language)’라는 데이터 타입은 말하고 듣는 음성과 쓰고 읽는 문자로 이루어져 있다. 예를 들어 영화 ‘아이언맨’을 친구에게 설명한다고 가정해 보자. 친구는 영화의 내용을 전혀 모르기 때문에 주어진 설명으로만 내용을 상상할 수 있다.

“억만장자 천재 발명가인 토니 스타크가 심장에 치명적인 상처를 입은 자신의 목숨을 지키며 아이언맨 강화 슈트를 제작한다. 세계를 지킬 과학의 결정체인 아이언맨은 범죄와 싸워 나간다.”

친구는 한 줄짜리 줄거리만으로 내용을 상상할 수 있을까. 아마 과학의 결정체인 강화 슈트라는 게 무엇인지부터 혼란을 겪을 것이다. 어쩌면 '슈트'라는 말로부터 턱시도 같은 의상을 생각할 수도 있다. 하지만 영화 포스터나 스틸컷을 보여줄 수 있다면 어떨까. 더 나아가 아예 영화를 보여줄 수 있다면 설명할 수고도 줄어들고 친구도 훨씬 더 깊게 이해할 수 있다.

백 번 듣는 것보다 한 번 보는 것이 낫다는 말도 있다. 영화라는 다차원 데이터(동영상)는 많은 정보를 포함할 수 있다. 반면 텍스트로 표현된 언어 데이터는 많은 정보를 굉장히 함축해 인코딩한 자료라는 특징이 있다.



'이해(Understanding)'한다는 것은 처리한다는 것보다 한 단계 더 높은 수준을 요구한다. 처리는 기계적으로 규칙을 따라 얼마든지 수행할 수 있지만 이해는 맥락 내용에 대한 파악을 전제로 한다. 보통은 자연어 처리(NLP)나 자연어 이해(NLU)라는 용어를 거의 구분 없이 사용하고 있지만 엄밀히 나누자면 자연어 처리에 해당하는 부분은 언어의 형식이나 구문론과 관련된 기능들로, 문장을 형태소 단위로 쪼갠다든지(POS), 구문 분석을 수행한다든지 하는 기능이 그 예다. 반면 자연어 이해는 문장 내 표현된 감성을 인식하는 감성 분류라든지, 문서 내 중요한 부분을 캐치하는 요약 등의 기능을 포함한다.

기계가 언어를 이해하기 위해서는 일단 언어를 인식해야 하는데 딥러닝을 적용하려면 인공 신경망에 언어 데이터를 넣어 모델을 학습시켜야 한다. 하지만 인공 신경망에 데이터를 적용하려면 데이터의 생김새가 인공 뉴런이 계산(가중 합이나 비선형 함수 적용)할 수 있는 형태여야 한다.

〈그림3〉

연휴는 시작하기 전이 가장 즐겁다

어절
 연휴는 + 시작하기 + 전이 + 가장 + 즐겁다

형태소
 연휴/NNG + 는/JX + 시작/NNG + 하/XSV + 기/ETN + 전/NNG + 이/JKS + 가장/MAG + 즐겁/VV + 다/E

음절
 연 + 휴 + 가 + 는 + 시 + 작 + 하 + 기 + 전 + 이 + 가 + 장 + 즐 + 겁 + 다

자소
 ㅇ + ㅓ + ㄴ + ㅎ + ㅠ + ㄴ + ㄴ + ㅡ + ㄴ + ㅅ + ㅏ + ㅣ + ㄴ + ㅈ + ㅓ + ㄱ + ㅎ + ㅋ + ...

여러 방식에 따른 문장 토크나이징 예

기계에 사람의 언어를 인식시키려면

텍스트로 된 언어는 계산 가능한 형태가 돼야만 인공 신경망으로 학습될 수 있다. 즉, 텍스트 데이터는 실수로 이뤄진 벡터나 매트릭스 등의 타입으로 변환돼야 한다. 이러한 다차원의 실수 데이터를 텐서(Tensor)라고 부른다. 이미지 데이터는 픽셀 값을 2차원 또는 3차원의 텐서로 만들어 인공 신경망에 태울 수 있다. 언어도 텐서로 바꿔 인식시킬 수 있다. 예를 들어 간단한 문장 하나를 예로 들어 보자.

토크나이징(Tokenizing) : 한 덩이로 돼 있는 문장을 인공 신경망에 인식시키기 위해서는 세부 단위로 쪼개는 작업이 필요하다. 이를 토크나이징 또는 파싱이라고 하며 이 쪼개진 단위를 토큰(token)이라고 부른다. 어떻게 쪼개면 좋을지는 언어의 특징과 수행하고자 하는 태스크와 데이터의 특징에 따라 달라진다.

일반적으로 한국어는 교착어라는 언어 특성상 문장을 형태소 단위로 자르거나 음절 단위로 자른다.

워드 임베딩(word embedding) : 한 덩이의 문장이 토큰들로 쪼개졌다면 이 토큰들을 인공 신경망이 계산할 수 있도록 벡터로 바꿔 줘야 하는데 이를 워드 임베딩이라고 부른다. 워드 임베딩 기법에는 여러 가지가 있지만 딥러닝에서는 사람의 언어 습득 과정과 유사한 워드 임베딩 학습 과정을 활용한다. 사람은 언어를 배울 때 사전을 찾아가면서 모든 단어의 정확한 의미를 습득하지 않는다. 비슷한 문맥에서 비슷한 위치에 등장하는 단어끼리는 유사한 의미를 가진

다는 것을 알게 되니 문장 내 어떤 단어의 의미를 만일 모른다고 해도 앞뒤 문맥이나 여러 예문을 보면 그 뜻을 대강 유추할 수 있다. 대표적인 알고리즘으로는 CBOW나 SKIPGRAM이 있다.

두 방식 모두 단어(토큰)를 특정 길이를 가진 임의 벡터로 만들어 주는데 많은 문장을 학습시키면 시킬수록 더 좋은 품질의 벡터가 나온다.

텍스트를 벡터로 바꾼 뒤라면 인공 신경망을 어떻게 구성하느냐에 따라 다양한 태스크를 수행할 수 있다.

문장·문서 분류(Sentence·Document Classification) : 입력받은 텍스트를 지정된 K개의 클래스(또는 카테고리) 중 하나로 분류하는 과제다. 인터넷 사용자 리뷰를 감성분석(긍·부정)하거나 사용자의 발화문을 챗봇이 처리할 수 있는 기능 중 하나로 매핑하는 의도 분류 등에 쓰일 수 있다.

[AI 이야기]

AI, 실제 현장 적용에 걸림돌 많아...

'전이 학습' 활용하면 소량 추가 학습만으로 성능 확보 가능

요구 사항 구체적으로 break-down 하기

AI 자동 회의 정리 솔루션

우선 스피커가 음성을 텍스트로 변환해야 하니까...상용 STT(Speech-to-text) 솔루션을 적용해야겠어!

변환된 텍스트 문장을 주제별로 묶어줄 수 있도록 토픽 클러스터링(Topic Clustering)을 추가해야겠군!

회의 기록은 구어체니까 회의록 형태로 바꾸려면 문서 요약 기술이 필요한데.. 아직 상용수진이 아니니 핵심 키워드 추출로 바꾸자!

이슈 감지를 하려면 우리 회사에서 자주 쓰이는 용어를 알아야 할텐데.. 다양한 주제로 최소 300시간 정도의 회의 내용이 학습 데이터로 필요할 것 같아!

AI가 이슈의 경중을 파악하는 기술은 어려우니 주제별로 정리된 회의 내용을 참고 자료로 주면 사람이 최종 컨펌해 주간 보고를 올리게 하자!

[한경비즈니스 칼럼=이주열 LG CNS AI빅데이터연구소장, 김명지 책임] 최신 기술을 소개하는 논문부터 모델 학습에 필요한 상세 코드까지 오픈 사이언스 인 딥러닝은 누구나 접근할 수 있는 곳에 무료로 모든 자료가 공개돼 있다. 글로벌 톱 티어 학회에 소개된 논문은 순식간에 전 세계의 연구자들에 의해 블로그에 쉽게 재해석되며 유튜브 영상으로 만들어지고 깃허브에 올라오는 공식 코드와 모델은 누구나 이용하기 편리하도록 2차, 3차 패키징이 되곤 한다. 온라인 커뮤니티는 최신 트렌드를 소개하고 다양한 기사와 의견, 자신만의 참신한 인공지능(AI) 프로젝트를 소개하는 사람들로 매일 붐빈다.

하지만 이 수많은 자료들은 쉽고 편하게 잘 다져진 튜토리얼에 불과하다. AI를 공부할 때는 무엇이든 다 할 수 있을 것 같았는데 막상 해보려고 하면 왜 잘 안 될까. 어느 기술 분야고 그렇지 않겠느냐마는 특히나 딥러닝 기반의 AI는 기술을 습득할 때와 실제 현장에 적용할 때의 괴리가 큰 분야 중 하나다. 딥러닝 모델이 실생활의 문제를 해결하기 위해 적용될 때도 잘 동작할 것이라는 보장은 없다. 현실에는 다음과 같은 걸림돌이 있기 때문이다.

구체적이지 않으며 불명확한 태스크

AI를 막 접하기 시작한 대부분의 사람들은 AI에 대해 크게 기대한다. AI가 우리 삶의 문제의 모든 것을 해결해 줄 수 있을 것이라고 막연히 생각하는 것이다. AI의 특장점이나 개념적인 부분, 포장된 사례만 접했다면 무리한 요구 사항을 내걸기도 한다. 분명 AI는 삶의 많은 부분을 더 편리하게 만들어 줄 수 있겠지만 모든 것에 대한 근본적인 해결책은 아니다. 두루뭉술하고 막연한 요구 사항보다는 구체적이고 명확한 목표를 세우고 필요한 하위 기능을 쪼개 생각해야 한다.

예를 들어 고객이 다음과 같은 요구 사항을 제시했다고 하자.

회의실에 AI 스피커가 하나 있어 자동으로 회의록을 작성한 다음 이슈 사항만 주간 보고로 취합해 줬으면 합니다

이 요구 사항은 다음과 같은 구체적인 하위 기능으로 쪼개 볼 수 있다.

대부분의 고객은 AI 전문가가 아니기 때문에 요구 사항을 기능 단위로 쪼개기 어려운 경우가 많다. 어떤 기능들이 AI로 가능한지, 그중 어떤 것은 현실적으로 아직 상용화하기에 무리가 있을지, 어떤 부분은 굳이 AI가 아니어도 처리할 수 있는지 등은 기술을 깊이 이해하고 적용해 본 경험이 없는 사람이라면 쉽게 도출하기 어려운 내용이다. AI 사이언티스트와 AI 엔지니어는 기술에 대한 이해를 바탕으로 상세 기능을 나누고 각 목표 수준을 정의한 다음 전체를 아우르는 시스템 구성을 애플리케이션 레벨로 설계할 수 있어야 한다.

작은 데이터, 낮은 품질의 데이터

태스크가 구체적으로 정해졌다면 AI 모델을 학습시킬 차례다. 단순하면서도 정해진 패턴을 다루는 태스크라면 상관이 없지만 이미지나 텍스트같은 비정형 데이터를 다루거나 코드로 판단·예측 규칙을 명문화하기 어려운 태스크라면 딥러닝 기반의 AI 모델을 이용하는 것이 좋다. 딥러닝 모델 학습은 사람이 규칙을 지정하지 않는 대신 데이터를 통해 패턴을 학습해야 하는데 그 학습 방법의 특성상 양질의 다양한 데이터를 필요로 한다. 하지만 대부분은 데이터를 차곡차곡 모아 놓은 뒤 AI 관련 사업을 발주하는 곳은 드물고 보유한 데이터가 많다고 해도 학습 데이터로 활용하기에 부적격인 것이 많다.

그렇다고 학습 데이터를 마련하는 것부터 시작하자니 이 또한 쉽지 않다. 요즈음은 AI 학습용 데이터를 라벨링하고 전처리해 주는 크라우드 소싱 업체도 많이 생겼지만 그래도 여전히 데이터의 소스를 제공해야 한다. 예를 들어 공정품의 양품·불량 체크를 위해서는 직접 공정품 사진을 찍어 제공해야 한다. 이마저도 다양한 데이터를 구축하려면 비용이 많이 들어갈 수도 있고 회사의 내부 데이터를 다뤄야 하다 보니 데이터의 반출이 불가능해 외부 업체를 활용하기도 어렵다. AI 모델 개발자는 늘 모자란 데이터와 싸워야 한다.

'전이 학습' 적용 사례

A 씨의 편의점 절도 감지 모델을 B 씨의 편의점에 적용할 때

영화 리뷰 평가에 대한 긍·부정 분류 모델을 뉴스 기사 댓글 긍·부정 분류에 적용할 때

C 회사의 e메일 보안 위반 탐지 모델을 D 회사에 적용할 때

일상생활의 다양한 이미지를 분류하는 모델을 농산품 품종 이미지 구분에 적용할 때

일반 상식에 대한 질의응답 모델을 금융권 콜센터 무인 자동 질의응답 과제에 적용할 때 등등

다른 도메인 환경

도메인 환경이 달라지는 것도 문제가 된다. 아무리 오버피팅을 피해 일반화한

모델이라고 해도 추론 환경이 달라지면 성능이 감소하기 마련이다.

동네에서 편의점을 운영 중인 A 씨는 아르바이트를 구하는 것이 어려워 야간 시간에는 무인 편의점으로 바꾸려고 한다. 이에 CCTV를 여러 대 설치해 계산하지 않고 나가는 손님이 발생하면 알림을 주도록 'AI 절도 감지 모델'을 만들기로 했다.

A 씨는 매장에서 다양한 절도 상황을 연출해 데이터를 만들고 모델을 학습시켰다. 감지 성능은 놀라웠다. 만득이의 모델은 한 달 동안 발생한 모든 절도 사건을 100% 검거했다. "이런 매장이 우리 가게뿐만이 아닐 텐데 이 모델을 다른 매장에 팔아 이용료를 받으면 나는 부자가 될 수 있겠군." A 씨는 해당 모델을 주변 편의점 점주들에게 팔았다. 하지만 결과는 참담했다. A 씨의 모델은 다른 편의점에서 전혀 절도를 감지하지 못했다. A 씨는 손해 배상을 하기 위해 자신의 매장을 내놓아야 했다.

A 씨가 간과한 것은 도메인 적응 문제(domain adaptation problem)다. 대부분의 딥러닝 모델은 동일한 기능을 수행하는 모델이라고 해도 추론 환경이 달라지면 제 기능을 수행하지 못한다. 다른 매장의 CCTV 화질, 밝기와 채도, 채광, 조명, 카메라 위치, 매장 내 크기, 선반 위치 등은 A 씨의 매장과 상이할 것이다. 이런 환경에서의 절도 행위는 모델이 학습한 적이 없기 때문에 제대로 검거하지 못하게 되는 것이다. A 씨의 편의점에서는 높은 검거율을 보였을지라도 이는 A 씨의 매장 환경에 한해서만 최적화된 성능이다. 다양한 편의점에서도 잘 작동하려면 다양한 편의점의 절도 데이터를 만들어 학습시켜야 한다. 또 다시 데이터 부족과 다양성 문제로 넘어가게 되는 것이다.

이처럼 하나의 AI 모델이 좋은 성능을 보인 전적이 있다고 해도 다른 환경에서 여전히 잘 작동한다는 보장을 할 수 없다. 열에 아홉은 데이터 부족으로 모델 학습에 고생하고 잘 만들어진 모델이라고 해도 또 다른 곳에서 잘될 것이라고 장담할 수도 없다. 성능 좋다는 AI 모델들은 다 이론적인 허상에 불과할까. 확장성 없이 매번 수만 건의 학습 데이터를 만드는 셋바퀴를 반복해야 하는 것일까.

전이 학습 : 한 번 만든 인공지능 모델 재사용하기

'전이 학습(transfer learning)'은 한 번 만들어진 딥러닝 모델을 재활용해 쓸 수 있는 기법이다. 비슷한 태스크를 다른 도메인에 적용할 때 그리고 그 태스크를 위한 학습 데이터가 부족한 경우 유용하게 쓰일 수 있다. 예를 들어 아래와 같은 다양한 경우 전이 학습을 적용해 볼 수 있다.

비슷한 경우에 대해 만들어 놓은 AI 모델이 있다면 다른 태스크 진행 시 밑바닥부터 모델을 만들 필요가 없다. 하지만 엄연히 환경이 다르고 데이터의 패턴이 다르기 때문에 그 모델을 그대로 활용해서는 좋은 성능이 나올 수 없다. 전이 학습은 하나의 모델이 이미 배워 놓은 지식을 잘 유지하면서도 새로운 태스크에 대해 필요한 지식을 추가로 습득할 수 있도록 한다. 이미 만들어 놓은 모델의 아키텍처를 새 태스크에 맞게 조금 수정하거나 추가하는 작업을 한 뒤 새로운 태스크에 대한 학습 데이터를 이어서 학습시킨다면 처음부터 학습한 모델보다 좋은 성능을 낼 수 있다. 새로운 학습 데이터가 기존만큼 많지 않은 환경에서도 말이다.

예를 들어 사과 농장의 의뢰를 받아 사과 사진으로 사과 품종을 자동 분류하는 과제를 수행한다고 가정하자. 하지만 확보한 품종별 사과 사진이 몇 장 없는 어려운 상황이다. 이대로 모델 A를 만들어 학습한다면 몇 장 없는 이미지에서 충분히 특징을 학습하지 못해 분류를 제대로 수행하지 못하게 된다. 하지만 약간 갈래는 다른 듯하지만 일반 식물의 종류를 구분하는 AI 모델 B를 만든 적은 있다. 만들어 둔 모델에 어떻게든 약간의 수정만으로 사과도 잘 분류하게 할 수 없을까.

우선 모델 B의 마지막 부분은 일반 식물을 분류하도록 돼 있을 테니 사과 품종 카테고리 수에 맞게 분류할 수 있도록 인공 신경망을 조금 수정해 줘야 한다. 이후 주어진 소량의 사과 이미지 데이터를 여기에 추가로 학습시켜 새로운 모델 B'를 만들어 낸다. 모델 B는 기존에 '식물 이미지' 데이터의 일반적인 특징(색상·각도·선·열매·잎사귀 등)을 학습한 적이 있으니 전이 학습을 적용한 모델 B'가 처음부터 소량의 사과만 학습한 모델 B보다 더 적은 학습량으로 더 많은 지식을 가질 수 있는 것이다.

치명적인 기억 상실(catastrophic forgetting)

태스크에 맞게 수정해 추가로 학습하면 모든 것이 좋아질 듯하지만 이것이 만능 해결책은 아니다. 딥러닝 모델이 새로운 정보를 학습할 때 이전에 배웠던 정보를 완전히 잊어버리는 경향이 발생하곤 하는데 이를 치명적인 기억 상실(catastrophic forgetting)이라고 한다. 사과 분류 모델의 예시로 돌아가 보자. 모델 B'가 몇 장 없는 사과 데이터에 대해 너무 여러 번 반복 학습을 하다 보면 모델 B가 이전에 학습했던 기본적인 식물 이미지의 특징들을 잊어버리게 될 수 있다. 전이(transfer)가 필요 이상으로 과하게 일어났다고 볼 수 있다. 치명적 기억 상실이 발생하지 않도록 하면서도 새로운 데이터를 잘 배우기 위해서는 전이 학습을 너무 과하게 시키지 않는 것이 좋다.

무인 편의점 결제 기술

동네에서 편의점을 운영 중인 A 씨는 아르바이트를 구하는 것이 어려워 야간 시간에는 무인 편의점으로 바꾸려고 한다. 이에 CCTV를 여러 대 설치해 계산하지 않고 나가는 손님이 발생하면 알림을 주도록 'AI 절도 감지 모델'을 만들기로 했다.




A 씨는 매장에서 다양한 절도 상황을 연출해 데이터를 만들고 모델을 학습시켰다. 감지 성능은 놀아웠다. 만득이의 모델은 한 달 동안 발생한 모든 절도 사건을 100% 검거했다. “이런 매장이 우리 가게 뿐이 아닐텐데 이 모델을 다른 매장에 팔아 이 용료를 받으면 나는 부자가 될 수 있겠군!” A 씨는 해당 모델을 주변 편의점 점주들에게 팔았다. 하지만 결과는 참담했다. A 씨의 모델은 다른 편의점에서 전혀 절도를 감지하지 못했다. A 씨는 손해 배상을 하기 위해 자신의 매장을 내놓아야 했다.

자료 : LG CNS

전이 학습 모델 이용

전이 학습은 보통 오픈 도메인 데이터에 대해 만들어 놓은 모델을 특정 도메인의 태스크에 적용하는 식으로 활용한다. 일반적인 내용을 두루두루 넓게 학습해 놓은 모델의 사전 지식을 구체적인 하위 영역에 활용하는 셈이다.

1) 컴퓨터 비전에서의 전이 학습

이미지를 입력 데이터로 처리하는 경우라면 이미지넷(ImageNet) 데이터로 학습된 모델에 전이 학습을 적용해 좋은 성능을 낼 수 있다. 이미지넷은 무려 120만 장의 학습 이미지를 1000개의 카테고리로 분류하는 과제이기 때문에 여기에 대해 학습한 모델은 이미지를 꽤나 잘 배웠다고 볼 수 있다. 특히 이미지넷 데이터 인식 대회에서 우승했던 GoogleNet(2014년 우승)이나 ResNet(2015년 우승) 등은 지금까지도 널리 활용되는 기본 모델이다.

이 밖에 다양한 특장점을 지닌 모델들의 아키텍처가 전부 논문으로 공개돼 있고 그중 대부분은 깃헙 등에 학습된 모델이 오픈도 있다. 이미지 처리를 하는 과제를 진행 중이라면 밑바닥부터 학습시키기보다 이런 자료를 활용해 전이 학습하는 것이 좋다.

ResNet의 구성

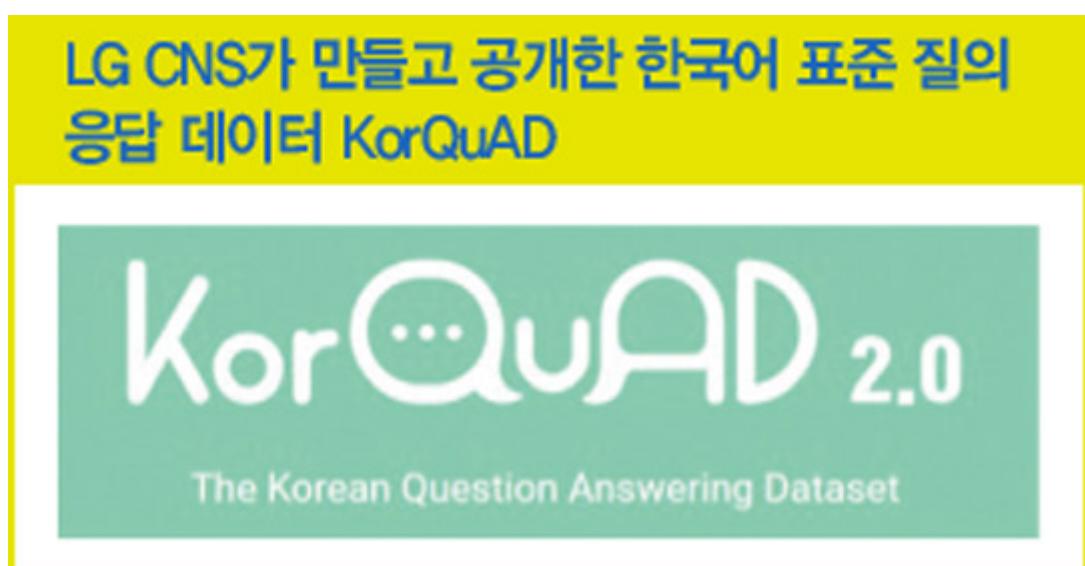




2) 자연어 이해(NLU)에서의 전이 학습

텍스트는 특히 함축적인 데이터 타입이기 때문에 전이 학습을 활용하는 것이 매우 효과적이다. 위키백과와 같은 방대하고 일반적인 지식에 대해서 학습한 적이 있는 모델이라면 일반적인 어휘의 의미를 대강 알고 있는 모델이라고 볼 수 있다. 이런 모델이 공개돼 있다면 전이 학습에 활용하는 것이 좋다.

대표적 자연어 이해 태스크인 자연어 질의응답은 AI가 수행하기 상당히 어렵고 복잡한 과제 중 하나다. 따라서 다량의 학습 데이터를 필요로 한다. LG CNS가 만들어 공개한 KorQuAD(Korean Question Answering Dataset)는 한국어 위키백과를 소스로 한 오픈 도메인 질의응답 데이터인데, 이러한 데이터로 학습해 둔 모델이 있다면 콜센터나 챗봇 등 특정 고객사의 질의응답 데이터가 부족해도 상대적으로 좋은 성능을 얻을 수 있다.



딥러닝은 데이터로부터 패턴을 학습해 의사 결정하는 모델이기 때문에 양질의 데이터가 충분한 환경이라면 아무 문제가 없다. 하지만 현실적으로 접근할 때는 데이터가 부족한 상황에 놓이는 경우가 대부분이다. 전이 학습은 과거 경험에 이어 소량의 데이터 추가 학습만으로 좋은 성능을 얻게 할 수 있는 AI 재활용 기법이다. AI 공부를 하려고 이론을 배우거나 튜토리얼 코드를 실습할 때 필

요한 것이 아니라 현장을 위한 기술이다. 그래서 카테고리 분류(classification)와 같이 어느 정도 기술 수준이 성숙된 과제에서는 전이 학습을 얼마나 잘 적용할 수 있느냐가 인기 있는 후속 연구 주제다. AI 모델을 제대로 학습시키기 어려운 만큼 한 번 만들어 놓았다면 알뜰살뜰하게 끝까지 잘 쓸 수 있어야 한다.

[본 기사는 한경비즈니스 제 1304호(2020.11.23 ~ 2020.11.29) 기사입니다.]

[AI 이야기]

무엇을 요청할지 모르니 미리 공부해 두는 방식...구글의 자가 지도 학습 모델 'BERT' 유용



- 1,000 object classes (categories).
- Images:
 - 1.2 M train
 - 100k test.



Classification task: produce a list of object categories present in image. 1000 categories.
"Top 5 error": rate at which the model does not output correct label in top 5 predictions

Other tasks include:

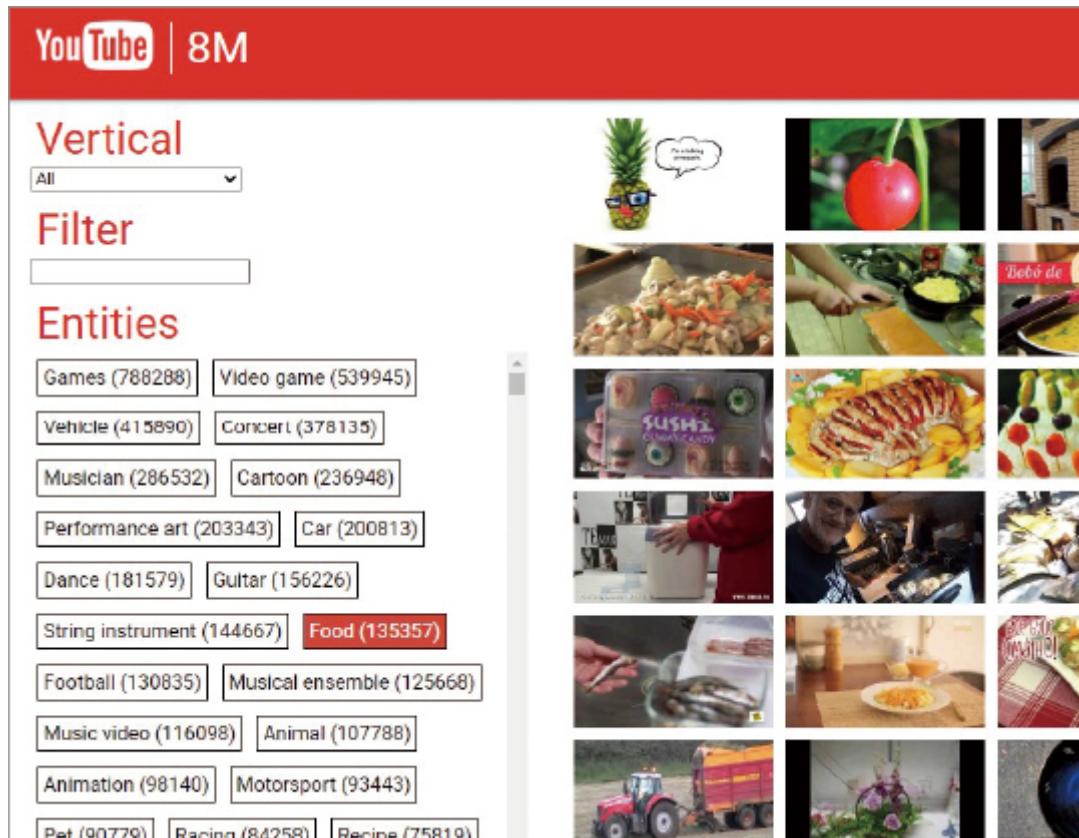
single-object localization, object detection from video/image, scene classification, scene parsing

[한경비즈니스 칼럼=이주열 LG CNS AI빅데이터연구소장, 김명지 AI빅데이터 연구소 책임] 전이 학습(transferring learning)을 해 보려고 해도 비슷한 태스크에 대해 만들어 놓은 딥러닝 모델이 없다면 재활용할 수 없다. 전혀 다른 것에 대해 학습한 모델이라면 전이 학습을 해도 효과가 없을 가능성이 매우 높다. 이는 마치 그림을 그리는 화가에게 갑자기 100m 달리기 경기에 나가 달라고 요구하는 것과 같아 모델 간의 사전 지식을 활용하기 어렵기 때문이다. 그렇다면 지식을 전수하는 전이 학습은 아주 유사한, 한정된 태스크 간에만 가능할까.

전혀 다른 태스크 간에는 지식을 전수하기 쉽지 않다. 하지만 전이 학습을 염두에 두고 다방면으로 활용할 수 있는 모델을 미리 만들어 놓을 수 있다. 여러 태스크에 활용하기 위해 여러 지식을 미리 두루두루 학습해 놓은 인공지능(AI)을 만드는 것인데 이러한 학습을 '사전 학습(pre-training)'이라고 하며 이러한 용도의 모델을 '사전 학습 모델'이라고 한다.

사전 학습은 보통 특정 데이터 타입에 대한 일반적인 지식을 두루 배워 놓는 것을 목표로 한다. 텍스트 사전 학습 모델은 언어의 일반적인 의미와 구조에

대해, 이미지 사전 학습 모델은 이미지의 일반적인 특징·색채·형태 등에 대해 배운다. 그래서 향후 어떤 텍스트나 이미지 관련 태스크를 수행한다고 하더라도 기본 지식을 바탕으로 잘 적용할 수 있게 된다. 마치 기본적인 웨이트 트레이닝·유산소운동·스트레칭 등을 통해 몸 쓰는 법을 배웠다면 어떤 운동 종목이든 적응하기 수월한 것처럼 말이다.



글문체 분류	매체 및 장르(1단계)	매체 부가정보(2단계)
현대문어	잡지(주간지, 월간지, 계간지)	경제, 과학, 국제, 기타(독자투고, 인물, 화제, 응어 등), 문화, 보도, 사회, 사회, 생활, 스포츠, 연극, 오픈미디어, 정치, 종교, 취미, 패션
현대문어	잡지(주간지, 월간지, 계간지)	교육자료, 사회, 상상적 텍스트, 생활, 예술론, 인문, 자연, 체험기술적 텍스트, 종류
현대문어	책	교육자료, 기타(독자투고, 인물, 화제, 응어 등), 사회, 상상적 텍스트, 생활, 수필, 예술, 예술론, 인문, 자연, 정보, 체험기술적 텍스트, 종류
현대문어	기타 출판물(만화문, 소설자, 청부문, 서 등)	예술론, 인문, 실기 르포
현대문어	화면이 있는 방송 녹화 전사	생활 대화, 사회 대화, 녹화/사회, 종류
현대구어	화면이 있는 방송 녹화 전사	사회, 생활, 예술론, 인문, 종류
현대문어	화면이 없는 방송 녹음 전사	예술론, 종류
현대구어	화면이 없는 방송 녹음 전사	사회, 생활, 신문, 예술론, 인문, 자연, 체험기술적 텍스트, 종류
현대문어	기타 녹음 전사	예술론
현대구어	기타 녹음 전사	-
현대문어	전자출판물	과학, 사회, 생활, 예술론, 인문, 자연, 체험기술적 텍스트, 도록, 종류

발음자 분류	폐체 및 장르(1단계)	폐체 부가정보(2단계)
현대문어	잡지(주간지, 월간지, 계간지)	경제, 과학, 국제, 기타(독자투고, 인물, 회제, 응어 등), 문화, 보도, 사설, 사회, 생활, 스포츠, 연극, 오피니언, 정치, 종류, 취미, 팬덤
현대문어	잡지(주간지, 월간지, 계간지)	교육자료, 사회, 산학적 텍스트, 생활, 예술론, 인문, 자연, 체험기술적 텍스트, 종류
현대문어	책	교육자료, 기타(독자투고, 인물, 회제, 응어 등), 사회, 산학적 텍스트, 생활, 수필, 예술, 예술론, 인문, 자연, 정보, 체험기술적 텍스트, 종류
현대문어	기타 출판물 (만화문, 소학자, 청부문, 서등)	예술론, 인문, 살기 브포
현대문어	화면이 있는 방송 녹화 전사	생활 대화, 사회 대화, 녹화/사회, 종류
현대구어	화면이 있는 방송 녹화 전사	사회, 생활, 예술론, 인문, 종류
현대문어	화면이 없는 방송 녹음 전사	예술론, 종류
현대구어	화면이 없는 방송 녹음 전사	사회, 생활, 신문, 예술론, 인문, 자연, 체험기술적 텍스트, 종류
현대문어	기타 녹음 전사	예술론
현대구어	기타 녹음 전사	-
현대문어	전자출판물	과학, 사회, 생활, 예술론, 인문, 자연, 체험기술적 텍스트, 토론, 종류

YouTube | 8M

Vertical

All

Filter

Entities

- Games (788288) Video game (539945)
- Vehicle (415890) Concert! (378135)
- Musician (286532) Cartoon (236948)
- Performance art (203343) Car (200813)
- Dance (181579) Guitar (156226)
- String instrument (144667) Food (135357)
- Football (130835) Musical ensemble (125668)
- Music video (116098) Animal (107788)
- Animation (98140) Motorsport (93443)
- Pet (90779) Racing (84258) Recipe (75819)

대규모 데이터에 대한 사전 학습

어떤 후속 태스크에 적용하든 잘 수행할 수 있도록 하려면 방대한 양의 지식을 골고루 배워 놓는 것이 좋다. 따라서 모델의 사전 학습은 대규모의 오픈 도메인 데이터에 대해 이뤄지는 것이 일반적이다. 이미지와 텍스트에 대한 대표적인 사전 학습 데이터와 태스크는 다음과 같다.

시각 데이터에 대한 사전 학습

많은 곳에서 언급되는 이미지넷 데이터 인식 대회는 가장 유명한 이미지 사전

학습 과제라고 할 수 있다. 120만 장의 학습용 이미지를 학습해 카테고리를 분류하는 이 과제는 대표적인 AI 이미지 인식 태스크다.

수많은 컬러 이미지를 1000개나 되는 카테고리로 분류하도록 학습된 모델은 일반적인 이미지의 특징 대부분을 다뤄 봤다고 봐도 무방하다. 이렇게 학습된 모델은 이미지에 대체로 어떤 색상들이 나타나는지, 등장하는 사물의 직선·곡선과 이들이 합쳐져 이루는 도형, 큰 개체와 작은 개체, 전경과 배경 등을 학습했을 것이다. 다양한 내용을 두루 살펴봤으니 이미지를 대상으로 하는 어떤 태스크에 적용된다고 해도 사전 지식을 기반으로 빠르게 학습할 수 있을 것이다.

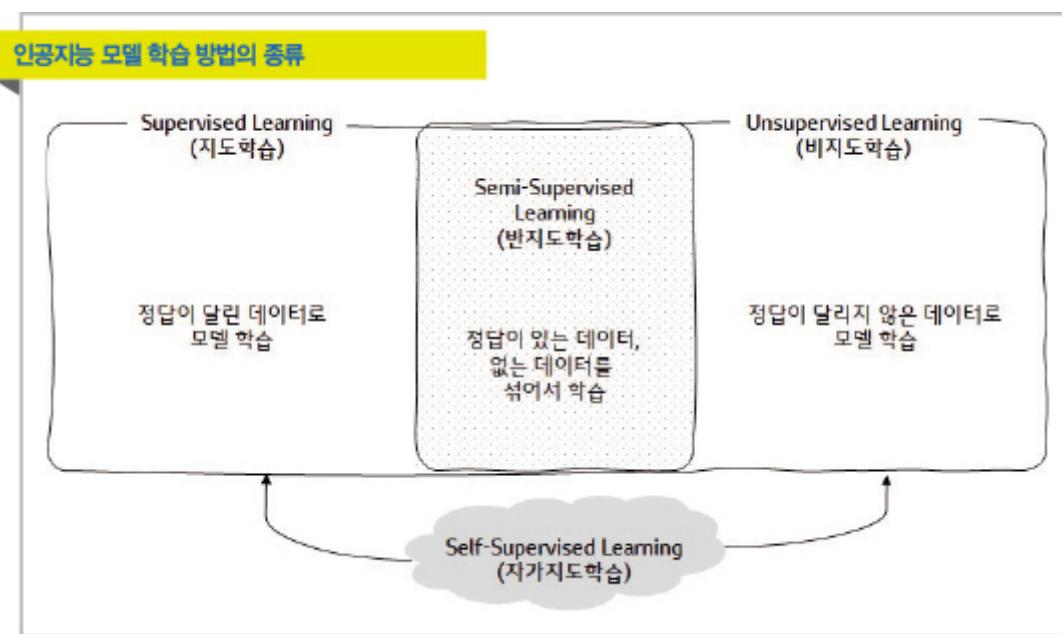
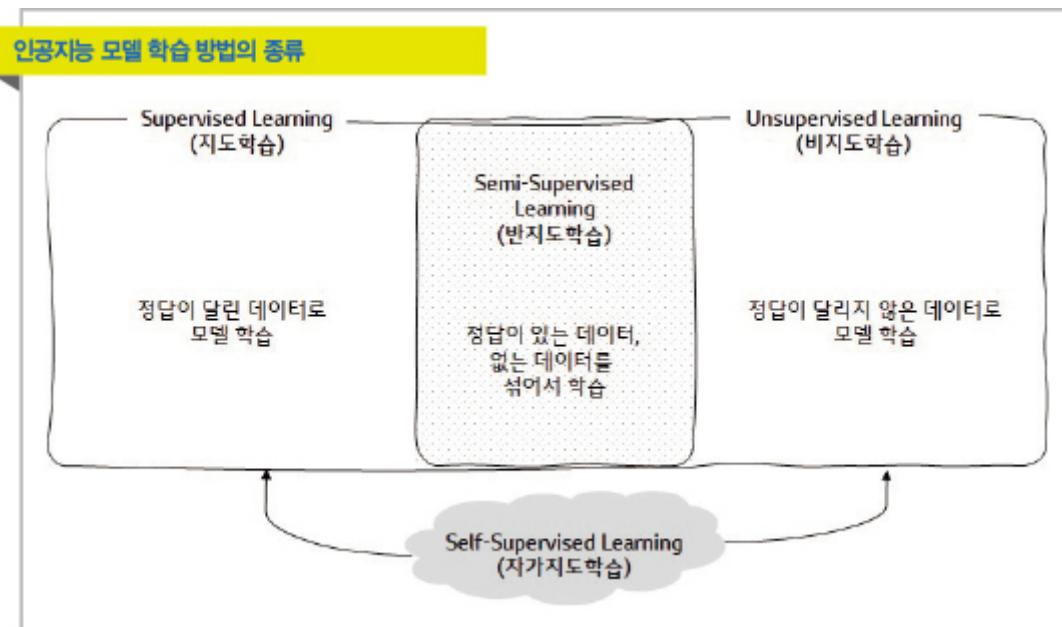
정지된 이미지뿐만 아니라 동영상에 대해서도 사전 학습용 데이터가 있다. 구글이 공개한 '유튜브-8M'은 무려 총 35만 시간에 달하는 610만 개의 비디오를 약 3800개의 카테고리로 다중 분류해야 하는 데이터다.

언어 데이터에 대한 사전 학습

언어에 대해 전반적으로 배워 놓는다는 것은 문맥에 따라 활용되는 단어의 의미·뉴앙스·적절한 문체 등을 습득한다는 것이다. 자연어는 이미지나 비디오 데이터와 달리 언어권의 차이로 인해 데이터를 언어권별로 각각 수집해 학습시켜야 하는 문제가 있다. 영어는 공개된 데이터가 많지만 한국어나 기타 언어에 대해서는 다양한 표준 데이터를 구하기가 쉽지 않다. 하지만 무난하게 활용하기 좋은 데이터로는 위키피디아가 있다.

위키피디아는 전 분야에 걸친 백과사전 지식을 대상으로 하기 때문에 텍스트 모델을 사전 학습시킬 수 있는 대표적 데이터다. 한국어로도 다운받을 수 있고 이외 여러 언어에 대해서도 제공하고 있다. 이 밖에 국립국어원에서 공개하는 세종말뭉치가 있다.

여러 매체로부터 모은 현대 문어·구어체를 제공하고 있어 경우에 따라 활용하기 좋다. 이 외에 뉴스·리뷰 등의 데이터도 자주 활용되는 사전 학습용 데이터다.



자가 지도 학습(self-supervised learning)

하지만 대규모로 구할 수 있는 데이터라고 해도 라벨까지 잘 달려 있는 것은 드물다. 라벨은 데이터에 대해 AI가 예측하기를 희망하는 결과다. AI는 입력 데이터를 받아 인식할 대상으로 여기고 라벨을 추론할 수 있도록 연산해야 한다. 입력 데이터만으로 학습할 수 있는 모델의 종류는 많지 않기 때문에 정답 라벨이 제대로 달려 있는 데이터를 얼마나 모을 수 있느냐가 모델 학습의 품질을 좌우한다. 하지만 정답 라벨링은 사람이 일일이 만들어 줘야 하기 때문에 공수가 많이 드는 작업이다. 어려운 일은 아니지만 인형의 눈알을 하나씩 붙이고 마늘을 하나씩 다듬는 것처럼 귀찮고 시간이 많이 드는 단순 반복 노동이다.

라벨을 붙인 데이터를 많이 만들기는 힘들어도 라벨 없는 데이터를 모으는 것

은 어렵지 않다. 수많은 이미지와 동영상·텍스트 문서 자체는 하루에도 셀 수 없는 양이 쏟아지고 있기 때문이다.

데이터가 많기는 한데 AI 모델에게 알려 줄 정답은 없고 어떻게 활용할 방법은 없을까.

이때 활용할 수 있는 학습 방법이 자가 지도 학습(self-supervised learning)이다.

'자가 지도 학습'이라는 용어를 처음 듣는다면 마치 AI가 스스로 학습해 똑똑해지는 것처럼 느껴지지만 그런 거창한 개념이 아니다. 자가 지도 학습은 사람이 만들어 주는 정답 라벨이 없어도 기계가 시스템적으로 자체 라벨을 만들어 사용하는 학습 방법이다. 사람이 라벨을 만들어 줄 필요가 없다는 점에서는 비지도 학습으로 볼 수 있지만 자체적으로 라벨을 만들어 사용한다는 점에서 지도 학습의 일종으로 볼 수도 있다. 다음은 자가 지도 학습의 예다.

예 : 이미지 데이터를 위한 자가 지도 학습

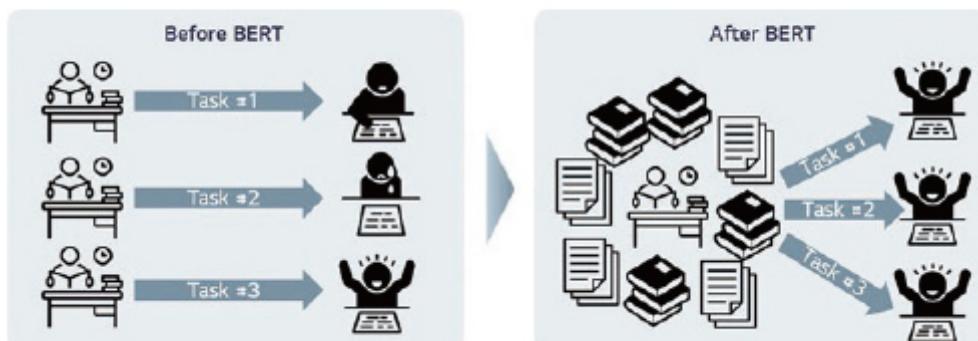
다람쥐와 청설모를 구별하고 싶은데 우선 종류에 상관없이 설치류 짐승의 사진을 10만 장 정도 충분히 많이 확보했다. 하지만 10만 장의 설치류 사진이 각각 어떤 종류에 해당하는지 라벨을 부여하기에 시간과 비용을 확보하기 어려웠다. 데이터는 많으니 우선 설치류의 일반적인 특징에 대해 조금이라도 알고 있는 딥러닝 모델을 사전 학습하려고 한다. 이때 자가 지도 학습을 활용해 자동으로 라벨을 부여하고 맞힐 수 있는 태스크를 만들어 모델을 사전 학습시키기로 했다.

이렇게 사전 학습한 모델을 다람쥐·청설모 데이터로 전이 학습했더니 다람쥐·청설모만으로 학습한 모델보다 좋은 성능을 얻을 수 있었다.

(왼쪽)BERT 이전의 자연어 처리 모델 학습 방식, (오른쪽)BERT의 학습 방식



(왼쪽)BERT 이전의 자연어 처리 모델 학습 방식, (오른쪽)BERT의 학습 방식



예 : 텍스트 데이터를 위한 자가 지도 학습

사외로 전송되는 e메일의 보안 위반 여부를 검출하려고 하는데 우선 보안 위반 여부와 관계없이 사외 전송 e메일 10만 건을 모았다. 하지만 10만 건의 e메일을 전부 살펴보기 힘들어 1만 건의 e메일에 대해서만 라벨링할 수 있었다. 가진 데이터를 전부 활용해 조금이라도 업무 관련 키워드를 학습할 수 있도록 자가 지도 학습으로 사전 학습시키고자 한다. 사전 학습 태스크로는 e메일의 중간 단어를 빈칸으로 대체한 후 들어갈 단어를 알아맞히도록 했다. 이렇게 하니 10만 건의 e메일을 전부 활용해 AI 모델에게 회사에서 자주 쓰는 키워드를 인식시킬 수 있었다.

이렇게 만든 모델을 전이 학습으로 활용하니 1만 건의 라벨링 데이터만으로 학습한 모델보다 좋은 성능을 보였다.

이처럼 자가 지도 학습은 주로 사전 학습에서 이용되며 다양한 데이터는 있지만 라벨이 없을 때 활용할 수 있다. 자가 지도 학습의 과제 자체가 의미 있는

것은 아니지만 수많은 데이터를 자체 라벨링으로 학습하게 되면 해당 데이터에 대한 전반적인 지식을 넓고 얕게 습득할 수 있게 되는 것이다. 이렇게 학습한 모델을 향후 후속 과제로 전이 학습하면 처음부터 특정 데이터로만 학습한 모델에 비해 일반적으로 좋은 성능을 보인다.

예 : 구글 BERT(Bidirectional Encoder Representations from Transformers)

자가 지도 학습 기법으로 사전 학습하고 다양한 태스크에 전이 학습할 수 있는 대표적인 예로 구글의 'BERT'라는 모델이 있다. AI에서의 자연어 처리는 BERT 이전과 이후로 나눌 수 있을 정도로, BERT는 자연어 처리 연구의 패러다임을 전환한 모델이다. 예전부터 사전 학습과 전이 학습의 개념이 있기는 했지만 기존의 사전 학습이 워드 임베딩 등 그저 보조적 역할을 수행하는 느낌이었다면 BERT는 사전 학습 자체가 주가 되는 모델이다. 예를 들어 기존 모델과 BERT의 학습 방식을 시험에 비유해 보자.

고3은 수능을 위해 수능 기출을 따로 풀고 토익 응시생은 토익 문제만 엄청 풀고 정보기술(IT) 회사의 사원은 정보 처리 기사를 공부해 각각의 시험에서 성적을 내기 위해 노력한다. 하지만 BERT의 관점은 '이것저것 잡히는 대로 책을 많이 본 사람'이 나중에 '어떤 시험을 치러도 잘 보게 된다'는 것이다.

즉 '언어'라는 분야 전반에 걸쳐 지식을 두루 쌓은 '하나의 거대한 뇌'를 사전 학습으로 만든다는 개념이다. BERT는 사전 학습에서 상당한 양의 데이터(텍스트 코퍼스)를 커다란 모델로 학습시켰고 후속 태스크를 위한 전이 학습은 간략하게만 진행해도 좋은 성능을 낼 수 있었다. 무려 11개의 자연어 처리 과제에서 1위를 차지했는데 이는 텍스트를 대상으로 할 수 있는 거의 대부분의 과제라고 볼 수 있다. 이때 BERT가 사전 학습한 문서가 무려 33억 단어만큼이고 16개의 TPUv3 칩을 활용해 학습했다.

어떻게 보면 당연한 결과다. 지식이라는 것은 서로 연결되는 부분이 있어 한 부분에서 습득했던 내용이 전혀 예기치 못한 다른 영역을 배우는 데 도움을 줄 수 있기 때문이다.

딥러닝 기반의 AI 모델은 다량의 양질 데이터를 필요로 한다. 이 중 대부분은 모델의 학습을 위해 사람이 태깅한 정답 라벨을 필요로 한다. 데이터 자체를 많이 확보하기는 쉬울지 몰라도 라벨링이 잘된 데이터를 다량으로 구하는 것은 쉬운 일이 아니다. 이때 활용할 수 있는 방법이 시스템적으로 라벨을 보유하고 학습할 수 있는 자가 지도 학습이다. 이 방식으로 모델은 전반적인 지식을 골고루 사전에 배워 놓을 수 있고 향후 특정 태스크로 전이 학습할 때 대체로 좋은 성능을 보인다.

일반적으로 자가 지도 학습을 활용한 사전 학습 모델은 다량의 방대한 지식을 골고루 습득하는 것을 목적으로 하기 때문에 대체로 모델의 사이즈가 큰 편이고 사전 학습 규모가 어마어마하다는 특징이 있다. 그래픽 처리 장치(GPU) 학습 장비나 데이터 저장 공간에 대한 비용 부담이 커 실용적이지 않게 느껴지기도 한다. 하지만 사전 학습 모델은 한 번 잘 마련해 놓으면 향후 어떤 과제든 적용할 수 있다. 장기적으로 두고 여러 곳에 활용할 수 있는 기초 모델을 준비한다는 개념으로 생각해야 한다.

[AI 이야기]

인공지능 제대로 가르치기...기계학습에서 모델의 일반화 성능을 높이는 방법

〈예시 1〉		
	A학생	B학생
5개년 과거 기출	상위 1% (전부 외워서 만점)	상위 10% (어려운 문제 몇 개 틀림)
올해 수능	상위 80% (기출 문제 그대로 나온 것 +찍은 것만 맞힘)	상위 10% (어려운 문제 몇 개 틀림)

[한경비즈니스 칼럼=이주열 LG CNS AI빅데이터연구소장, 김명지 AI빅데이터 연구소 책임]

참고 자료

Michael Nielsen, Neural Networks and Deep Learning,
<http://neuralnetworksanddeeplearning.com/>

회사원들이 우스갯소리로 말하는 '시연의 법칙'이라는 것이 있다. 분명 어젯밤에 테스트해 볼 때는 잘 작동하던 것이 중요한 보고나 고객 앞에서 시연할 때만 갑자기 기능이 동작하지 않고 느려지고 예외가 발생하는 일이 있다.

오버피팅과 일반화 성능

억울한 일이지만 이러한 문제는 기계 학습에서도 굉장히 자주 발생한다. 학습 시킬 때는 주어진 데이터를 잘 맞혔는데 이상하게도 운영 환경에만 올리면 추론 성능이 똑똑 떨어지곤 한다. 이는 기계 학습에서 가장 중요한 개념 중 하나인 일반화 성능과 연관이 있다. 일반화 성능은 '이전에 본 적 없는 데이터'에 대해서도 잘 수행하는 능력이다. 즉, 우리가 만든 인공지능(AI) 모델은 훈련 시에는 본 적이 없는 새로운 입력 데이터에 대해서도 잘 수행돼야 한다는 것이다. 훈련 시에만 잘 작동하고 일반화 성능이 떨어지는 모델을 오버피팅(overfitting)됐다고 한다.

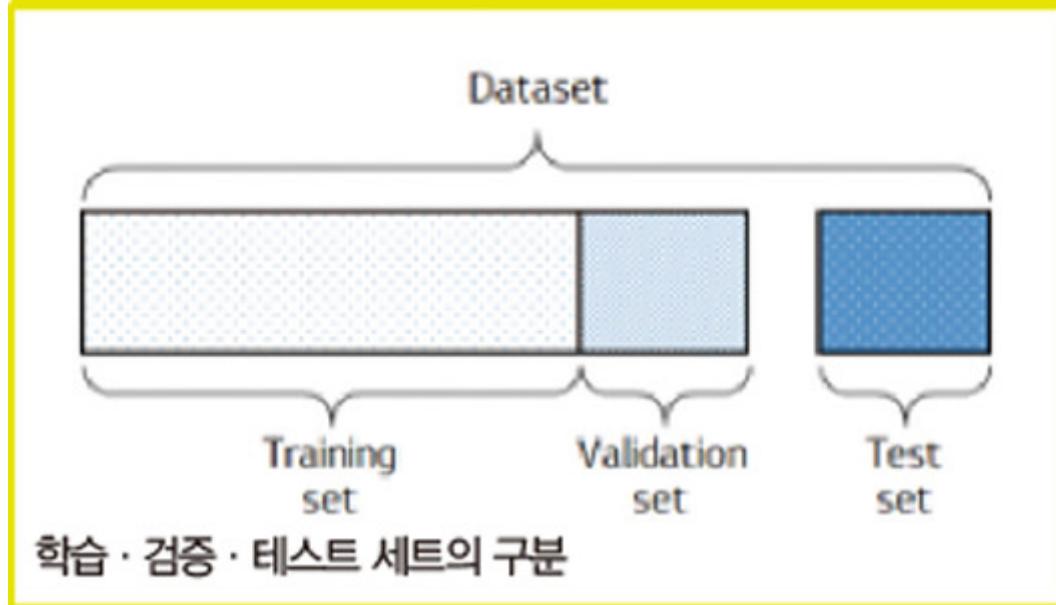
예시를 통해 알아보자.

에서 A학생과 B학생은 고3 수험생으로 올해 수능을 치른다. 남은 시간은 한 달 남짓이다. A학생은 이해력이 조금 떨리지만 굉장히 노력파로, 남은 한 달 동안 최근 5년간의 수능 기출을 전부 암기해 버렸다. 그래서 A학생은 문제만 들으면 정답이 몇 번 보기에 어떤 문장이었는지 정확하게 복원해 냈다. 다만 문제의 의도가 무엇인지, 왜 그 보기와 정답인지는 이해할 수 없었다. 단순히 문제와 답을 외웠기 때문이다.

B학생은 한 달간 과거 기출의 출제 의도를 파악하고 그 보기가 왜 정답인지 이해하며 공부했다. 최근 5개년의 출제 동향이나 문항 패턴을 파악했지만 어려운 문제도 있어 기출 문제를 전부 맞힐 수는 없었다. 기출을 암기한 A학생과 내용을 이해한 B학생은 올해 수능 수험장에 들어갔다. 수능 결과가 어땠을까. 여기서 A학생은 AI 모델로 따지자면 오버피팅된 모델이라고 볼 수 있다. 이미 습득한 데이터에 대해서는 모두 외웠기 때문에 기가 막히게 잘 맞힐 수 있지만 일반화 능력이 떨어지기 때문에 외운 것과 다른 데이터에 대해서는 제대로 역할을 수행하지 못한다.

만일 AI를 도입하려는 적용처가 늘 한정된 데이터 몇 가지만 다루는 곳이라면 데이터를 외워 추론하는 모델을 만들어도 무관하다. 이때 머신러닝이나 딥러닝 모델을 사용하지 않는다고 해도 간단하게 처리할 수 있는 것이 많을 것이다. 하지만 현장은 다양한 패턴의 데이터가 셀 수 없이 쏟아져 나오는 경우가 대부분이다. 특히 사진·동영상·글·음성 등과 같은 비정형 데이터를 다루는 곳이라면 더더욱 그러하다. 미리 쌓아 놓은 데이터를 잘 맞히도록 학습하는 것은 물론 중요하지만 앞으로 실시간으로 발생할 데이터도 잘 추론하도록 하는 것이 목적이 돼야 한다. 즉, 운영 환경에서의 일반화 성능 또한 학습 과정에서의 최적화만큼이나 중요하고 늘 이를 염두에 두고 모델을 만들어야 한다.

〈그림1〉



학습·검증·테스트

학습된 모델이 현장에서도 잘 작동할지는 어떻게 확인할 수 있을까. 먼저 확보한 데이터를 학습·검증·테스트의 3세트로 나누고 각 세트가 수행할 역할을 구분해 주도록 한다.

각 데이터 세트의 역할은 다음과 같다.

1) 학습 세트(training set)

학습 세트는 머신러닝·딥러닝 모델을 학습하는 데 이용하는 데이터다. 모델은 학습 세트의 입력 데이터와 정답을 보고 정답을 더 잘 맞히기 위해 노력한다. 이는 단순 최적화에 해당한다.

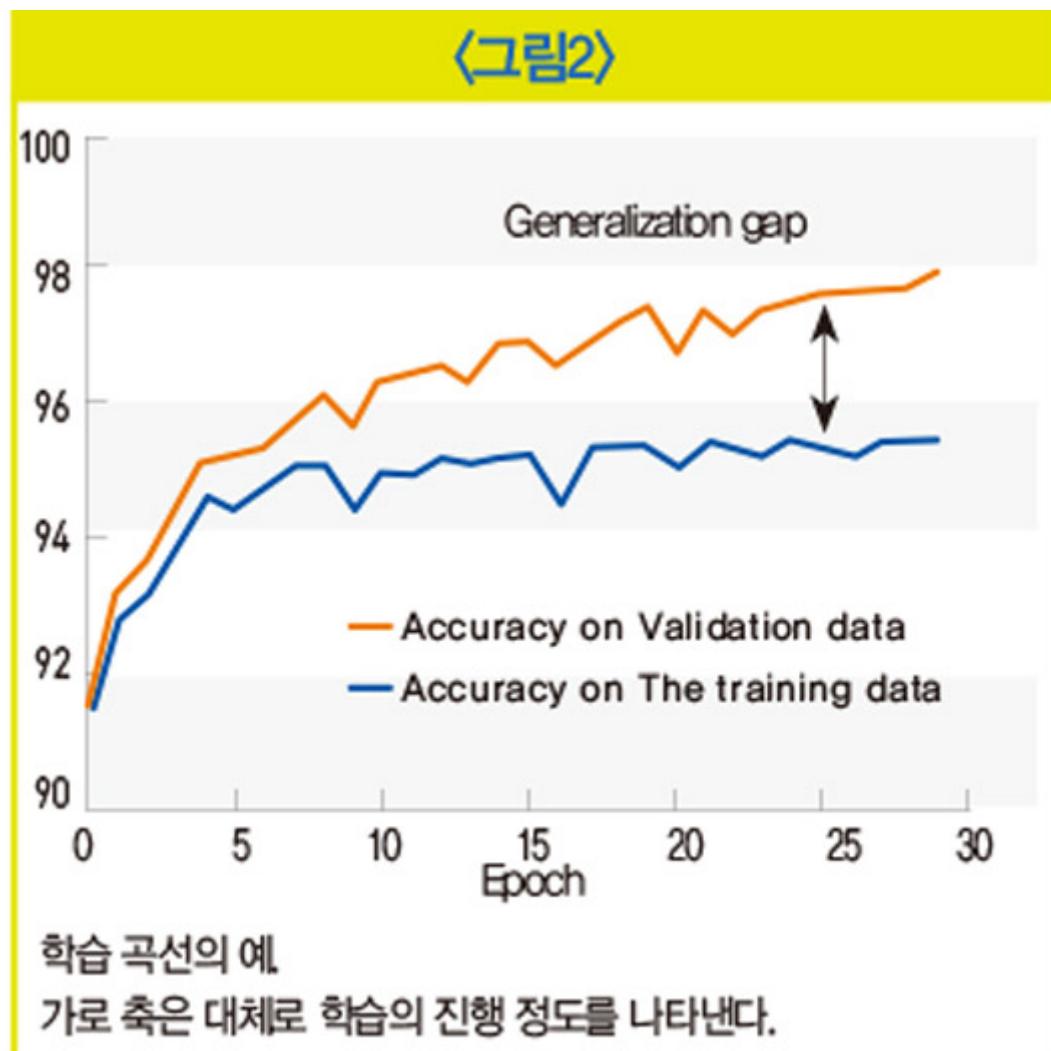
2) 검증 세트(validation set)

검증 세트는 머신러닝·딥러닝 모델에 정답을 알려 줄 데이터는 아니지만 모델을 튜닝하는 데 도움을 주는 데이터다. 즉 모델의 일반화 성능을 판단해 이어질 실험을 계획하는 데 이용한다. 모델은 검증 세트의 정답은 본 적이 없지만 이 세트의 입력 데이터만으로 일단 정답을 추론하게 된다. 모델이 배우지 않았던 데이터, 즉 처음 보는 데이터에 대해 얼마나 잘 맞히는지 계산할 수 있으니 이 결과를 보고 우리는 실험을 개선할 수 있다. 예를 들어 우리가 만든 모델이 학습 세트에 대해서는 잘 맞히는데 검증 세트에 대해서는 너무 못 맞힌다고 가

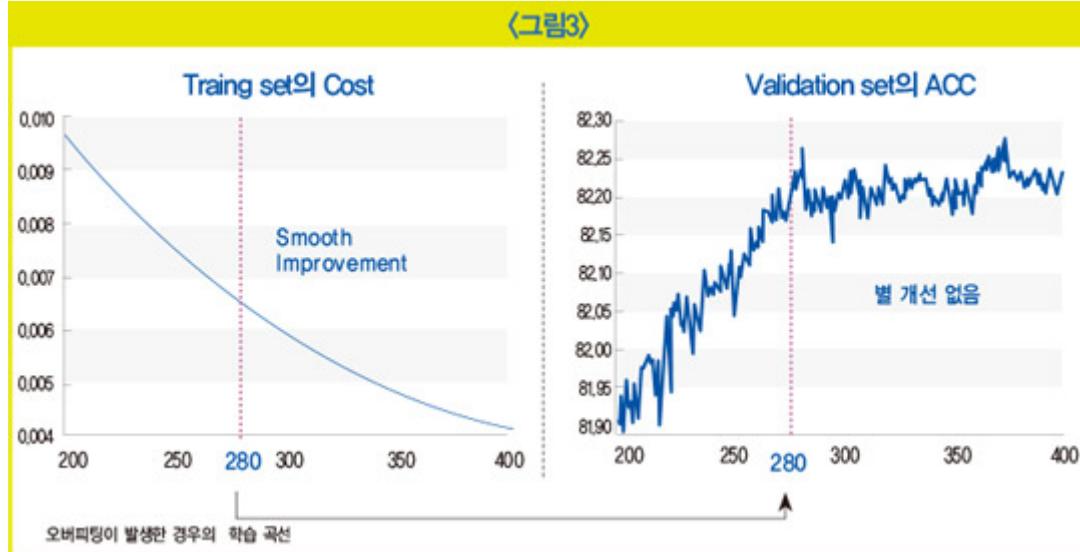
정하자. 위 예제의 학생A처럼 알려준 것만을 달달 외우고 일반화 성능이 부족한 오버피팅 현상이 발생했다고 볼 수 있다. 이를 통해 다음 실험 때는 오버피팅을 방지할 전략을 세워볼 수 있다.

3) 테스트 세트(test set)

테스트 세트는 모델의 학습에 어떤 식으로도 전혀 관여하지 않는 데이터로, 오로지 모델의 최종 성능을 평가하기 위해 따로 떼어 놓은 데이터다. 여러 모델 간 성능을 비교할 때 테스트 세트에 대한 스코어를 활용한다. 학습 세트로 모델을 학습하고 검증 세트에 대한 성능을 확인하며 모델을 개선해 왔으니 그 결과를公正히 평가하기 위한 기준이다. 학생A와 학생B가 치른 금년도 수능 시험과 같은 데이터라고 볼 수 있다. 모든 수험생이 제각기 다른 방식으로 공부했겠지만 모두 처음 보는 문제로 동시에公正하게 평가하는 기준이 수능이다.



〈그림3〉



학습 곡선 확인하기

역할에 따라 데이터를 나눴다면 학습이 제대로 이뤄지는지, 일반화 성능이 떨어져 오버피팅이 발생하지 않았는지 확인해야 한다. 확인하는 방법은 학습 곡선을 그려보는 것인데, 학습 곡선은 학습이 진행됨에 따라 모델의 성능을 기록하는 그래프다.

오버피팅이 발생했는지 확인할 수 있는 방법은 학습 곡선상에서 학습 세트와 검증 세트에 대한 모델의 성능이 어떻게 변화하는지를 확인해 보는 것이다. 정답을 알면서 배우는 데이터에 대해서는 당연히 잘 맞혀야겠지만 정답을 모르고 추론만 진행하는 데이터에 대해서는 어떤 경향을 보이는지 살펴보는 것이다. 어느 순간 검증 세트에 대해서 성능 개선이 없다면 모델은 '학습'이 아닌 '암기'를 하는 것이라고 볼 수 있다. 학습 곡선은 전형적인 오버피팅의 예다.

위 은 400 에포트 동안 모델 학습을 시켰다. 왼쪽의 그래프를 보면 학습 세트, 즉 모델에게 정답을 알려주면서 잘 맞히도록 유도하는 경우에 대해서는 문제 없이 최적화가 진행되는 것처럼 보인다. 왼쪽 그래프의 세로축을 이루는 '코스트(Cost)'라는 것은 실제 정답과 모델이 예측한 예측 값의 차이를 정량화한 수치로, 작을수록 모델이 잘 맞힌 것이다. 하지만 오른쪽의 그래프를 보면 모델이 정답을 모른 채 예측만 해야 하는 검증 세트에 대해서는 280 에포크 즈음부터 크게 개선된 게 없다는 것을 볼 수 있는데 이때의 세로축은 분류 정확도 (accuracy)로, 그 값이 클수록 분류를 잘한다는 뜻이다. 학습 세트에 대해서만 성능을 개선하고 있고 검증 세트에 대해서는 별다른 향상이 없는 시점이 오버피팅 발생 시점이다.

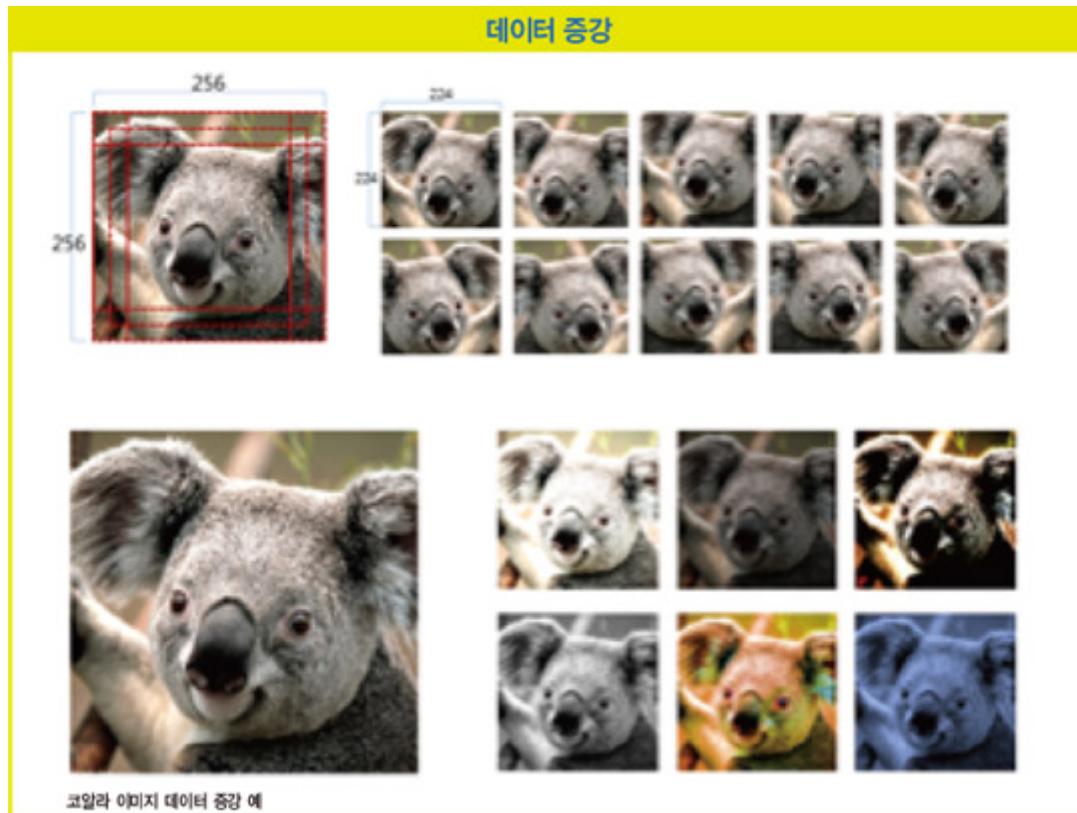
정규화

만일 머신러닝·딥러닝 모델을 학습할 때 과 같은 경향을 마주했다면 다음 실험에서는 이 현상을 개선하기 위한 전략을 취해야 한다. 오버피팅을 피하기 위한 모든 전략들을 정규화(regularization)라고 한다. 정규화의 목적은 일반화 성능을 향상하는 데 있다. 학습 세트에 대해 더 잘 맞히고자 하는 것이 아니라 현장에 나가 처음 보는 데이터에 대해서도 잘 맞힐 수 있도록 하는 목적이다. 이를 위해 취할 수 있는 대표적인 정규화 방법을 간단히 소개한다.

4) 데이터 증강

오버피팅을 피하는 가장 좋은 방법은 데이터를 더 많이 확보하는 것이다. 다양한 데이터가 있다는 것을 알려준다면 AI 모델도 현장에 나갔을 때 새로운 데이터를 더 잘 맞힐 수 있을 것이다. 하지만 현실적으로 데이터가 절대적으로 부족한 경우가 대부분인데 이를 위해 부족한 데이터 수를 많은 것처럼 증강시키는 다양한 기법이 있다.

이미지 데이터 학습을 예로 든다면, 한 장의 이미지를 좌우로 반전시키거나 일부 영역을 크롭하거나 노이즈를 추가하거나 색상·명암·채도 등에 변화를 줘 모델 학습에 추가로 데이터를 이용할 수 있다. 이렇게 하면 AI에 더 다양한 환경에서 찍은 듯한 이미지를 학습시키는 효과를 주어 오버피팅을 방지할 수 있다. 하지만 너무 과도한 변형은 오히려 해가 될 수 있으니 주의해 이용해야 한다.



5) 모델 수용력 줄이기

모델 수용력은 모델의 복잡한 정도를 나타내는 개념이다. 일반적으로 딥러닝 모델이 머신러닝 모델보다 모델 수용력이 높고 그중에서도 신경망을 여러 층 쌓거나 뉴런의 수를 많이 둘수록 수용력이 높아진다고 말할 수 있다. 수용력이 높은 모델은 처리할 데이터의 복잡 다양한 패턴을 더 잘 담아낼 수 있다.

하지만 수용력이 필요 이상으로 너무 높은 모델은 주어진 데이터를 외우게 될 가능성이 높다. 따라서 수행하려는 태스크에 알맞은 수용력을 가진 모델을 선택하는 것이 좋은데 태스크의 복잡도와 수용력의 관계는 딱 정해진 규칙이 없어 어느 수준이 적정선이라고 말하기는 어렵다. 만일 오버피팅의 경향이 발견된다 싶으면 모델의 층수를 줄여본다든지 한 층의 뉴런 수를 줄인다든지 등의 조치를 취해 볼 필요가 있다.

6) 조기 종료

조기 종료라는 기법은 말 그대로 오버피팅이 감지될 경우 목표하는 학습 시간이 다 되지 않았다고 하더라도 '조기 종료'해 버리는 것이다. 학습 곡선 예제 그림의 경우 400에포크를 학습하기로 계획했지만 오버피팅이 감지되니 280 에포크쯤에서 학습을 중단할 수 있다. 요즈음의 기계 학습 프레임워크에서는 조

기 종료를 적용하기 쉽도록 관련 기능을 잘 패키징해 둔 경우가 많다.

7) 드롭 아웃

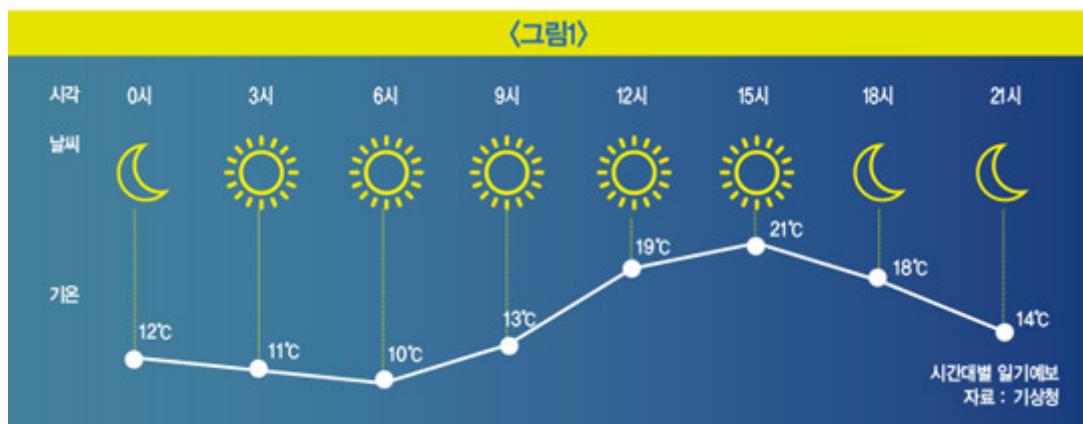
드롭 아웃은 학습 과정에서 일정 비율만큼의 노드를 무작위로 끄고 학습을 진행하는 기법이다. 딥러닝 모델 학습 시 상당히 많이 적용하는 정규화 기법 중 하나인데, 일부 노드가 사라진 상태에서 남아 있는 노드만으로 어떻게든 정답을 맞혀야 하는 딥러닝 모델은 훨씬 더 강력해질 수 있다. 없어진 주변 노드의 몫까지 수행해야 하니 훨씬 기능이 강화된 모델이 되는 것이다.

AI를 현장에 적용할 때도 '나무를 보지 말고 숲을 보라'는 말이 유효하다. 이미 확보해 놓은 과거의 데이터에 대해 좋은 성능을 낼 수 있도록 잘 학습하는 것, 이는 물론 굉장히 중요하지만 목적은 거기에서 끝나지 않는다. 잘 만들어진 모델이 향후 실제 현장에서도 좋은 성능을 내며 데이터를 처리할 수 있게 되는 것이 최종 목표일 것이다.

기계 학습에서 한 번에 완벽한 모델이 만들어지는 경우는 없다. 첫째 모델의 성능이 엉망진창이라고 해도 실패한 것은 아니다. 다만 어떤 현상이 문제인지 확인하고 앞으로의 개선 전략을 잘 세워 둘째·셋째 모델을 만들 수 있으면 된다.

[AI 이야기]

누적된 시계열 정보를 미래 예측에 활용...연산 과부하 줄이려고 중요 데이터만 '기억'하기도



[한경비즈니스 칼럼=이주열 LG CNS AI빅데이터연구소장·김명지 AI빅데이터연구소 책임] 시계열 예측 분석은 대표적인 데이터 분석 과제다. 이는 순차적인 시계열 데이터를 활용해 근미래에 어떤 데이터 값이 나타날지 예측하는 과정으로, 시간 흐름에 따라 변화하는 로그 데이터를 활용한다. 주가 예측이나 기상 예측이 그 예다. 시간대별로 기온을 예측해야 하는 기상청을 예로 들어보자. 다음 시각의 기상 정보를 예측하려고 할 때 한 시각의 입력 데이터만 고려하지 않을 것이다.

시간 흐름에 따른 데이터 처리하기

현 시각의 기온을 예측할 때 이전 시각의 기온, 바람의 풍속, 습도 데이터가 중요하다고 가정하자. 일기 예보관은 세 시간마다 이전 시각까지 확보한 데이터를 가지고 현 시각의 기온이 어떻게 될지 예측할 것이다,

9시엔 이전 시각의 기온 10도, 풍속 시속 29km, 습도 80%라는 값을 활용해 9시의 기온이 13도가 될 것이라고 예측한다. 12시에는 9시의 기온(13도), 풍속과 습도를 가지고 19도가 될 것이라고 예측하고 15시에도 마찬가지 작업을 한다. 매 시각마다 예측하고자 하는 Y값(기온)을 맞추기 위해 이전의 기온·풍속·습도라는 입력 값 X를 활용하는 것이다. 여기까지는 기존의 인공지능(AI) 학습 방식과 동일하다. 입력 값 X를 가지고 Y를 예측한다는 것은 이미지를 넣으면 강아지와 고양이를 구별한다든가 텍스트 문장을 넣어 긍정과 부정을 분류하는

문제와 같다.

여기서 한 아이디어가 머릿속에 떠오른다. 일기 예보관은 매시간 같은 일을 반복하는데 일을 반복하면서 쌓인 예측 노하우가 있을 것이다. 또한 밤낮이라는 시간의 흐름, 계절이라는 흐름에 따른 패턴이 분명히 존재할 텐데 지금처럼 해당 시각의 입력 값 3개만으로 기온을 예측하는 것은 활용할 수 있는 정보를 다 쓰지 못하는 것처럼 보인다. 한 시각의 데이터 세 개 만으로 예측할 수도 있겠지만 과거의 정보들이 연속된 형태니까 어떤 식으로든 현 시각의 예측에 도움이 될 것이다.

〈그림2〉				
시각	9시	12시	15시	18시
이전시각 기온	10	13	19	21
이전시각 풍속	29	32	25	18
이전시각 습도	80	90	70	70
현시각 기온	13	19	21	?

세 시간마다 기온 예측

순환 신경망 RNN

이런 형태의 데이터를 잘 다루기 위한 인공신경망 종류가 있다. '순환 신경망'이라고 하는 RNN(Recurrent Neural Network)이다.

기존의 신경망이 벡터 또는 매트릭스로 변환된 입력 데이터를 가지고 출력한다면 RNN 역시 동일 작업을 하지만 하나 차이점이 있다. RNN은 데이터를 처리해 결과를 출력했던 과거 과정의 일부를 가져와 현시점에서 데이터를 처리하고 결과를 출력하는 데 도움을 주는데 에서 오른쪽을 향하는 가로 방향의 화살표에 해당한다. 이때 벡터 형태로 누적된 과거의 정보(feature)가 넘어간다. 그래서 둘째 시점에는 해당 시각의 입력 데이터뿐만 아니라 첫째 시점의 누적된 정보를 포함해 예측하게 된다. 셋째 시점에는 첫째와 둘째 시점의 누적된 정보가 반영될 것이고 넷째 시점이 되면 세 시점의 압축된 정보가 도움을 준

다. 입력 데이터만으로 예측하는 것보다 과거의 누적된 정보까지 활용한다면 더 개선된 예측을 할 수 있다.

RNN의 특징은 다음과 같다.

장점1 : RNN은 시간 흐름에 따른 과거 정보를 누적할 수 있다. 기상청 기온 예측 사례와 마찬가지로 RNN이라는 특수한 형태의 신경망은 입력 데이터뿐만 아니라 과거의 처리 내역을 반영해 더 나은 결정을 할 수 있다는 것이 가장 큰 장점이다.

장점2 : RNN은 가변 길이의 데이터를 처리할 수 있다. 기온 예측 사례의 경우 3시각 단위로 예측하는데 이러한 단위 시간마다의 매 시점을 타임스텝(timestep)이라고 한다. 타임스텝은 시간 단위가 될 수도 있고 일·월·초 단위 등 순서만 있다면 구성할 수 있다. RNN은 과거의 정보를 매 타임스텝마다 압축해 다음 타임스텝으로 넘기므로 데이터의 길이에 무관하게 자유롭게 구성할 수 있다. 하루 치 데이터로 예측할 수도 있고 1주일 치, 한 달 치 데이터를 모아 예측할 수도 있다.

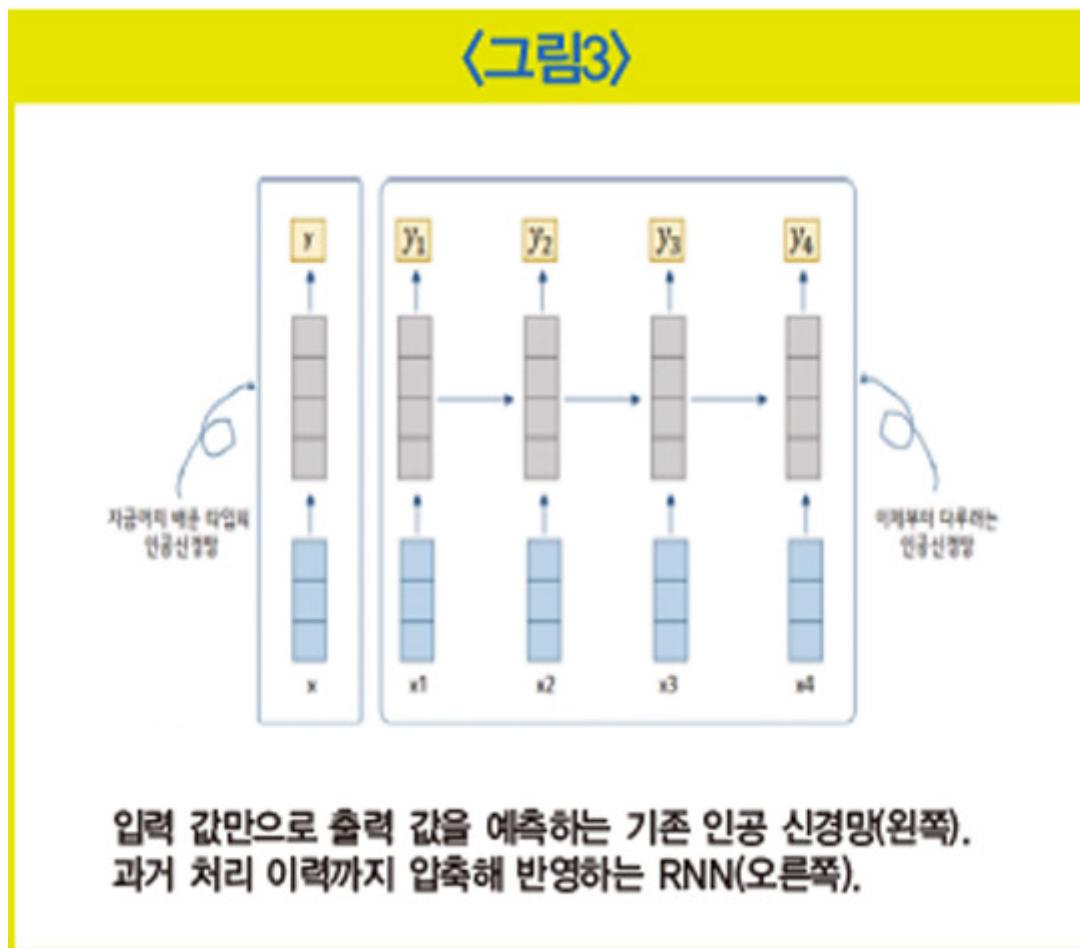
장점3 : RNN은 다양한 구성의 모델을 만들 수 있다. 유연한 구조를 가진 RNN은 상황에 따라 다양하게 입력 데이터를 처리, 누적하고 결과를 예측하도록 신경망을 구성할 수 있다.

이때 입력 데이터의 정보를 누적하는 부분을 인코딩(encoding), 결과를 출력하는 부분을 디코딩(decoding)이라고 한다.

이때의 단점은 연산 속도가 느리다는 것이다. RNN은 과거의 처리 내역을 현재에 반영해야 하기 때문에 현시점의 데이터를 처리하려면 반드시 이전 시점의 데이터가 처리 완료돼야 한다. 따라서 병렬 학습이 어렵고 순차적으로 데이터를 처리해야 하는 성질 때문에 연산 속도가 다소 느린 편이다. 딥러닝 모델 학습 시 많은 경우 그래픽처리장치(GPU) 서버를 활용하곤 하는데 RNN을 학습할 때 GPU 병렬 처리의 이점을 잘 활용할 수 없는 한계가 있다. 정형 데이터(수치·범주형 등)를 예측할 때 속도 저하를 크게 체감하지 못하는 것이 대부분이지만 텍스트 데이터를 다룰 때는 종종 느린 속도가 문제가 되곤 한다.

또한 실질적으로 과거 정보를 잘 활용할 수 있는 모델은 아니다. 이론적으로 RNN은 과거의 정보가 누적되며 현재 추론에 도움을 주지만 실질적으로 먼 과거의 정보를 반영하기는 힘들다. RNN은 한 타임스텝씩 정보를 누적해 인코딩하는데 먼 과거의 정보는 여러 번 압축되고 누적되다 보니 거의 영향을 미치지 못한다. 이를 RNN의 장기 종속성 문제(long-term dependency)라고 한다.

이는 사람도 마찬가지다. 오늘 점심 메뉴는 뭐였는지 기억하지만 어제 점심이 뭐였는지, 1주일 전 점심은 무엇이었는지 바로 기억할 수 있는 사람은 많지 않다. 최근의 정보일수록 잘 기억하고 반영하며 먼 과거일수록 잊어버리는 경향이 인공 신경망에서도 동일하게 나타난다.

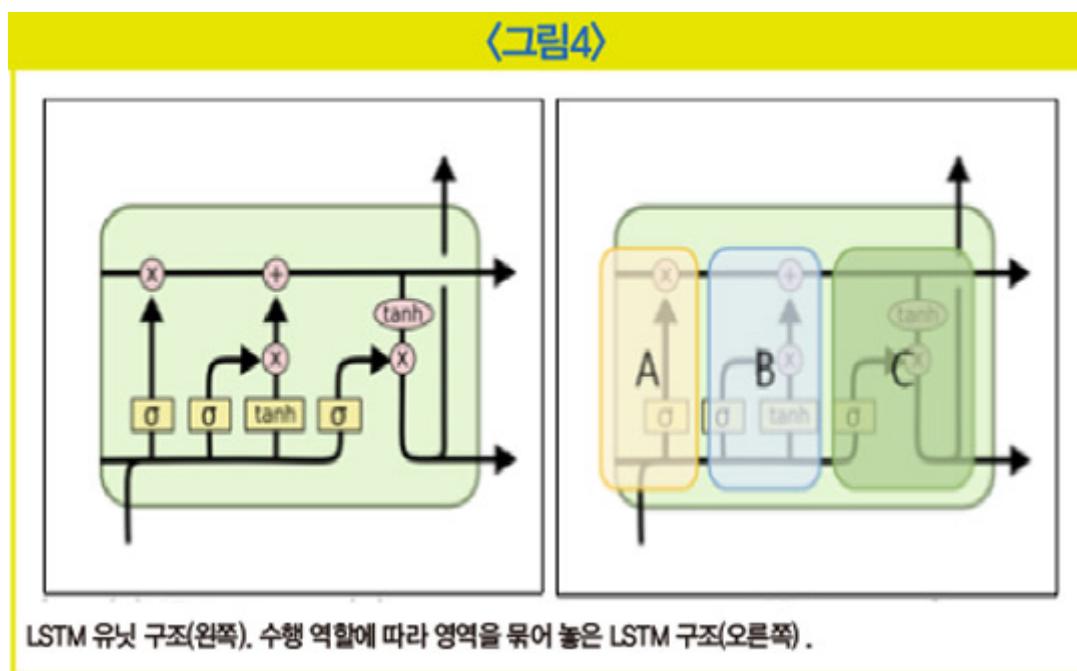


RNN의 성능을 보완한 LSTM

RNN의 이러한 단점을 해결하는 방안도 나왔다. 먼 과거 시점의 정보 중 중요한 것은 기억하고 불필요한 것은 잊어 버리도록 스스로 조절할 수 있는 RNN 유닛을 만드는 것이다. 대표적으로 LSTM(Long-Short Term Memory) 유닛이 있다. 장·단기 메모리 유닛이라는 뜻인데 아래와 같은 구조이다.

LSTM에는 세 부분의 게이트(gate)가 있는데 정보의 흐름을 조절하는 관문 역 할을 수행한다. 의 A 부분은 망각 게이트(forget gate)라고 불리며 말 그대로 잊어버림에 대해 조절한다. 과거의 정보 중 불필요하다고 생각하는 부분은 망각하기로 결정한다. B 부분은 입력 게이트(input gate)로, 현재의 정보(input data)를 얼마나 반영할지 결정한다. C 부분은 출력 게이트(output gate)로, 현 시점에 연산된 최종 정보를 다음 시점에 얼마나 넘길지 결정한다. 이 세 게이트는 정보의 흐름을 AI가 자체적으로 더 원활하게 조절하는 기능을 한다. 따라서 데이터가 길어진다고 해도 기본 RNN에 비해 더 좋은 예측을 할 수 있다.

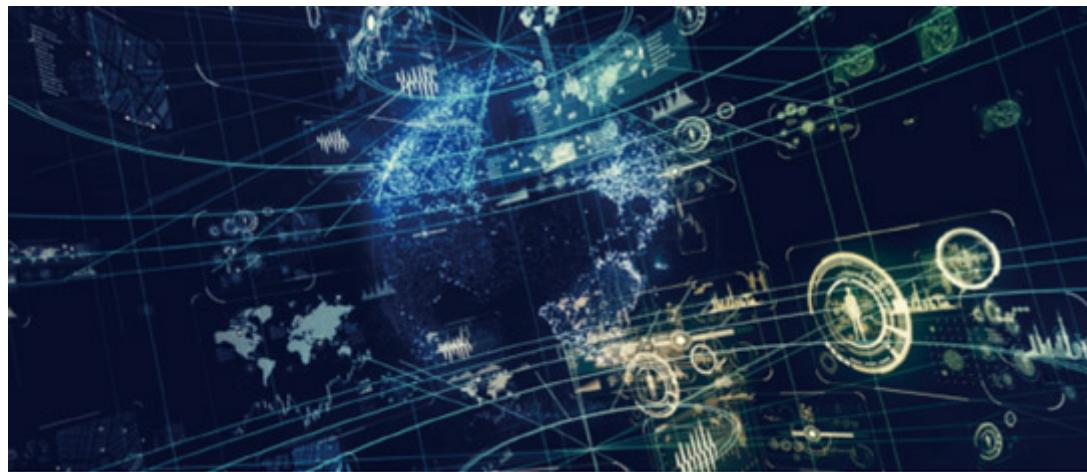
오늘날 인공 신경망을 활용하는 대부분의 시계열 예측은 아주 간단한 과제를 제외하고는 LSTM과 같은 개선된 유닛을 활용한다. 대부분의 딥러닝 학습용 프레임워크에서 이를 쉽게 구현할 수 있도록 기능을 제공하고 있다.



사람은 오래전부터 과거의 경험을 통해 미래를 예견하고자 했다. 사람은 한 번 수행했던 업무라면 노하우를 얻어 다음에 더 빠르고 잘 수행할 수 있고 과거에 실패를 경험했다면 같은 실패를 다시 경험할 가능성도 낮다. 인간의 신경계를 본뜬 인공 신경망에도 비슷한 기능을 수행하는 알고리즘이 있다. 인공 신경망에서는 RNN이라고 불리는 특별한 형태의 신경망이 그 역할을 수행한다. AI가 미래를 예측한다는 것은 미래 예언의 개념이 아니다. 과거의 패턴을 통해 근미래에 벌어질 가능성이 가장 높은 데이터를 수리 통계적으로 유추하는 것이다.

[HELLO AI] LG CNS의 AI 이야기

-이미지 파일은 2차원 픽셀 행렬 데이터로 구성돼...인공신경망 통해 특징 추출하는 연산 수행

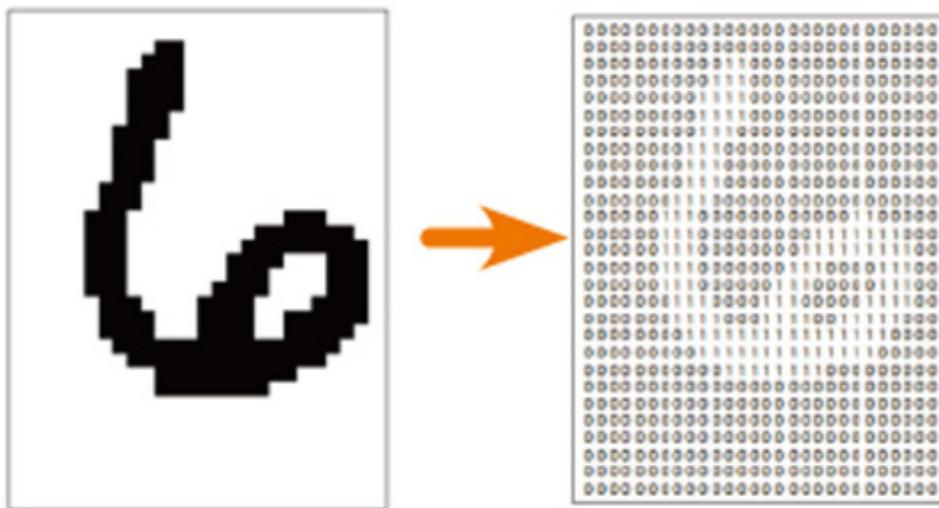


[이주열 LG CNS AI빅데이터연구소장
김명지 AI빅데이터연구소 책임]

'시각'은 동물이 가진 가장 원초적 감각 중 하나다. 동물들은 태어나 눈을 뜬 그 순간부터 잠들지 않는 동안 늘 눈으로 들어오는 각종 시각 정보를 인식해 판단하고 행동한다. 사람은 무의식적으로 아주 빠르게 시각 정보를 처리할 수 있도록 진화했다. 그렇게 해야만 더 잘 살아남을 수 있었기 때문이었다. 예를 들어 초식동물인지 맹수인지, 나무 넝쿨인지 뱀인지, 독버섯인지 식용버섯인지를 구별해야 하는 등 시각은 생존과 직결된 문제였다. 시각이란 사람에게는 매우 익숙한 감각이지만 기계도 과연 시각 정보를 인식할 수 있을까.

이미지와 같은 비정형 데이터를 다루는 데 특화된 딥러닝 모델은 인공 뉴런을 여럿 연결한 인공 신경망을 기반으로 한다. 인공 신경망은 가중합(weighted summation)과 비선형 함수(non-linear function)로 이뤄진 연산을 수행해야 한다. 따라서 입력 데이터로 벡터나 행렬과 같은 형태를 필요로 한다. 그렇다면 색상·곡선·각도·도형·명암이 어우러진 이미지 데이터를 어떻게 처리해야 인공 신경망에 입력할 수 있을까.

〈그림1〉



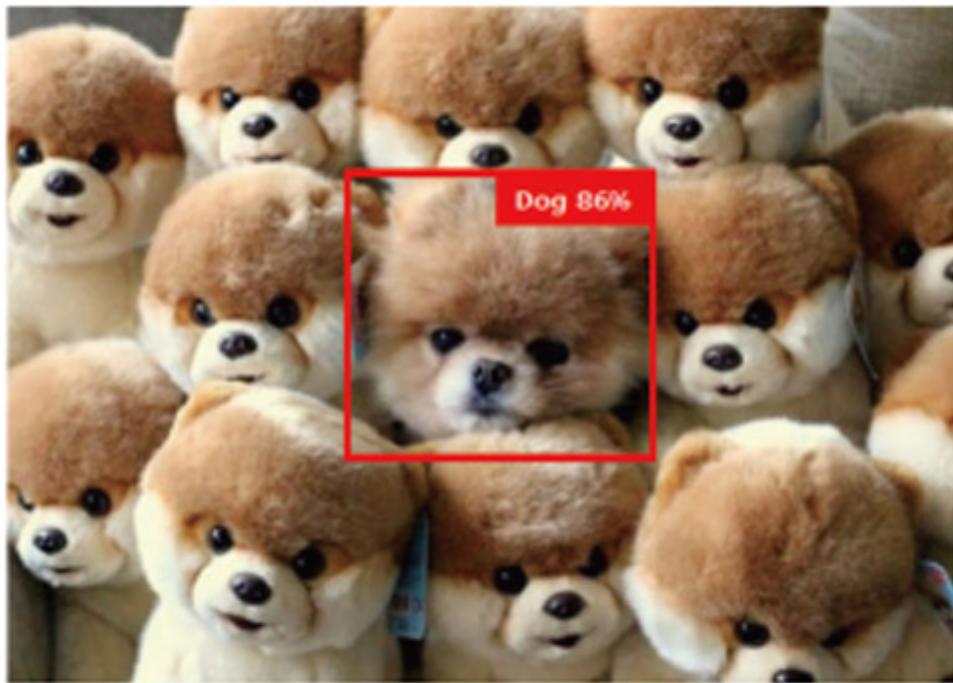
숫자 흑백 이미지 데이터, 사람이 보는 이미지(왼쪽),
컴퓨터가 인식하는 이미지(오른쪽)

〈그림2〉



이미지 분류 예.
강아지 사진을 입력받아 견종을 구별하는 AI 모델.

〈그림3〉



객체 탐지 예. 사진 내 강아지를 탐지하는 AI 모델.
구글 검색 이미지.

〈그림4〉

셀 값에 대해 가중한 연산을 수행한다

1 <small>x_1</small>	1 <small>x_0</small>	1 <small>x_1</small>	0	0
0 <small>x_0</small>	1 <small>x_1</small>	1 <small>x_0</small>	1	0
0 <small>x_1</small>	0 <small>x_0</small>	1 <small>x_1</small>	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

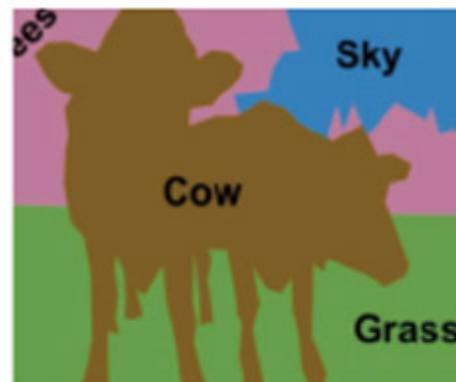
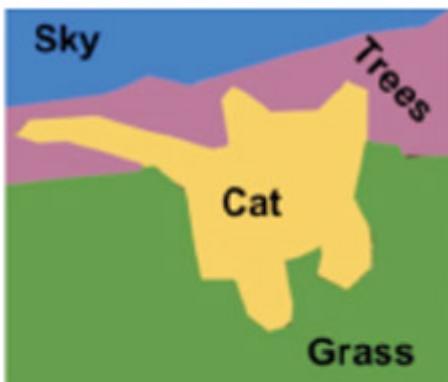
Convolved Feature

〈그림5〉



판다 사진 원본(좌) 노이즈 낸 판다 사진(우)

〈그림6〉



이미지 영역 분할 예. 사진상에 등장하는 다양한 객체의 영역을 구별하는 모델. 자료 : cs231n

기계가 이미지를 인식하는 방법

JPG와 PNG 파일 등으로 구성된 이미지 데이터는 우리의 눈으로 보기에는 하나의 정지된 그림이지만 기계는 이를 연산 가능한 형태로 만들어 인식한다.

예를 들어 온 사람이 보기엔 숫자 6이지만 기계가 보기에는 가로와 세로 28픽셀씩을 갖는 2차원 행렬 데이터다.

이미지를 행렬로 변환했기 때문에 이제는 인공 신경망 연산을 할 수 있게 된다. 이때 가로세로의 공간적 정보를 담고 있는 이미지를 처리하기 위한 특별한 인공신경망 구성 방식이 필요하다.

보통은 CNN(Convolutional Neural Network)이라고 불리는 형태의 인공신경망을 활용해 이미지에서 주요 특징을 추출하는 연산을 수행한다.

CNN은 이미지의 픽셀 값을 가지고 추출한 특징으로부터 그림의 숫자가 무엇인지 인식해야 한다.

특징의 추출은 컨볼루션 필터(또는 커널)가 수행한다.

이 필터는 이미지를 상하좌우로 훑으며 픽셀 값에 대해 가중합 연산을 수행한다.

이렇게 찾아낸 특징은 이미지의 색상이나 테두리, 각도와 같은 정보를 포함한다. 컨볼루션 연산을 여러 번 반복하면 인공신경망은 점차 추상적이고 복합적인 특징을 찾아낼 수 있다. 신경망의 깊이가 깊어질수록 인공지능은 강아지와 고양이를 분류하거나, 사람의 얼굴을 인식하거나, 제품 공정에서의 불량품을 탐지하는 등의 고도화된 역할을 수행할 수 있다.

컨볼루션 필터가 이미지를 상하좌우로 훑으면 특징을 찾아내면 이 결과로부터 정보를 추리는 풀링 연산이 이어진다. 컨볼루션 필터가 추출한 특징을 상하좌우로 훑으며 핵심적인 정보만을 영역별로 샘플링하는데, 주로 영역 내 가장 큰 값만을 남기고 나머지 값을 버리는 맥스풀링(Max Pooling) 방식을 적용한다.

컨볼루션 연산이 이미지의 특징을 찾아낸다면 풀링 연산은 그중 핵심 정보만 남긴다. 대부분의 이미지 처리 모델에서는 컨볼루션과 풀링 연산을 여러 번 반복하면서 데이터의 특징을 추려낸다. 이미지로부터 특징을 배워 나가는 작업이라는 뜻에서 이 과정을 피처러닝(feature learning)이라고 부른다.

대표적인 인공지능 이미지 인식 과제

인공 신경망이 복잡한 연산을 통해 이미지에서 주요 특징을 찾아냈다면 이 정보를 활용해 목표로 하는 태스크를 수행할 수 있다. 오늘날 인공지능(AI)이 이미지 데이터를 처리하는 대표적인 태스크를 몇 가지 소개한다.

1. 이미지 분류(image classification)

입력으로 받은 이미지를 지정된 K개의 클래스(또는 카테고리) 중 하나로 분류하는 과제다. 기계가 사진 속 동물이 강아지나 고양이인지 분류한다거나 공장에서 AI가 카메라로 벨트 컨베이어의 제품을 찍으며 사진을 보고 양호한 상태인지 특징인지 판별하는 업무 등에 쓰일 수 있다.

2. 객체 탐지(object detection)

입력 받은 이미지에서 특정 개체가 어디에 위치하는지 픽셀의 (x, y) 좌표 값을 찾아주는 과제다. 대부분의 스마트폰에는 인물 사진을 찍을 때 자동으로 인물의 얼굴 부근에 네모 박스를 찾아 포커스하는 기능이 있다. 이 경우 얼굴을 탐지하는 기능이 탑재돼 있다고 볼 수 있다.

3. 이미지 영역 분할(image segmentation)

객체 탐지가 네모 박스로 개체의 좌표를 찾아준다면 이미지 분할은 조금 더 정밀하게 픽셀 단위로 영역을 구별하는 과제다. 경계를 잘 찾아야 하는 작업인 만큼 객체 탐지보다 기계가 수행하기 어려운 태스크다.

이 밖에 이미지를 처리하는 다양한 AI 기술이 있고 나날이 기술이 빠르게 발전하고 있다. 특히 대표적인 글로벌 이미지 분류 대회인 국제영상인식대회 (ILSVRC : ImageNet Large-Scale Vision Recognition Challenge)에서는 이미 5년 전 AI의 수준이 사람의 인식률을 뛰어넘기도 했다. 이는 2012년 최초의 '딥러닝' 기반 모델이 이 대회에서 우승하며 딥러닝의 전성기를 가져온 지 3년 만의 일이다.

적어도 이미지 분류에서는 이제 AI가 사람보다 인식을 더 잘한다고 볼 수 있다.

아직까지 AI가 사람처럼 복합적인 추론 과제를 수행할 수는 없지만 단순한 수학 연산을 엮은 구조만으로 기계가 이 정도로 이미지를 인식할 수 있다는 것은 놀랍다. 어떤 관점에서는 0과 1의 디지털 데이터만 다루던 기계에 세상을 볼 수 있는 '눈'이 생긴 것이라고 볼 수도 있다.

기술이 조금 더 발전한다면 사람처럼 시각 인지를 바탕으로 행동하거나 다양한 사고를 추론하는 것도 가능해지지 않을까.

[본 기사는 한경비즈니스 제 1295호(2020.09.19 ~ 2020.09.25) 기사입니다.]

[AI 이야기]

학습 데이터 라벨링에 막대한 자금·시간 소요...뇌 MRI 영상은 전문의 작업 필요

〈표1〉 능동 학습(active learning) : 족집게 데이터로 공부하기

철수의 학습 계획	철수는 문제집 5권을 임의로 뽑아 안에 있는 문제를 전부 풀었다.
영희의 학습 계획	영희는 문제집 3권을 임의로 뽑아 안에 있는 문제를 전부 풀었다. 영희는 풀었던 문제집 3권에서 많이 틀리는 유형들을 체크했다. 그 뒤 남은 7권의 문제집에서 많이 틀리는 유형에 해당하는 200문제를 추가로 찾아 풀었다.

[환경비즈니스 칼럼=이주열 LG CNS AI빅데이터연구소장, 김명지 책임] 능동 학습(active learning)이라는 기술은 데이터가 많지만 인공지능(AI)을 '학습시킬 데이터'를 마련하기 쉽지 않을 때 이용할 수 있는 기술이다. 강아지 이미지와 고양이 이미지를 구별하는 태스크라면 누구나 데이터를 라벨링(labeling)할 수 있어 적은 비용으로도 금방 데이터를 모으겠지만 뇌 자기 공명 영상 장치(MRI)로부터 파킨슨 병 여부를 판단하는 태스크라면 해당 분야의 전문 의사가 아니라면 몹시 어려운 일일 것이다. 게다가 이미지를 한 장씩 보면서 파킨슨 병 여부를 일일이 태깅할 만큼 시간 여유가 있는 뇌 전문의를 구한다는 것 또한 쉬운 일은 아니다. 하지만 뇌 전문의 다섯 명이 한 달 동안 매일 20장씩만 이미지를 보고 라벨링을 해줄 수 있다면 어떨까. 수많은 뇌 MRI 영상 중 가능한 한 파킨슨 병인지 아닌지 가려내는 데 효과적인 데이터만 뽑아 정답을 알려 달라고 하고 싶지 않을까.

능동 학습은 라벨링을 할 수 있는 인적 자원은 있지만 많은 수의 라벨링을 수행할 수 없을 때 효과적으로 라벨링을 하기 위한 기법이다. 수행하고자 하는 태스크가 너무 특수해 해당 도메인의 전문 인력만이 데이터를 라벨링할 수밖에 없다면 최대한 학습에 효과적인 데이터만 뽑아내는 데 쓰일 수 있다. 뇌 MRI 영상으로부터 파킨슨 병을 진단할 수 있을 만한 대표적인 특징이 있을 것이다. 하지만 누구나 파악할 수 있는 대표적인 특징을 가진 데이터 말고 너무 모호해 전문의가 아니고서는 판단하기 어려운 데이터를 확보할 수 있다면 AI 학습에 도움이 되지 않을까.

수학능력시험을 예로 들어보자. 수능일까지 남은 시간 동안 각 수험생이 풀어 볼 수 있는 문제의 수는 한정돼 있다. 고3 수험생 철수와 영희가 있다고 생각해 보자. 철수와 영희는 각각 동일한 문제집 10권을 가지고 있는데 문제집 1권당 문제가 100개씩 있다. 철수와 영희는 이 중 총 500문제를 풀고 난 후 수능을 보게 된다.

철수와 영희의 학습 방법이 과 같을 때 누가 더 효과적으로 학습했을까. 일반적으로 본다면 영희가 더 효과적으로 학습했다고 말할 수 있다. 풀 수 있는 문제가 500개로 제한된 상황이라면 많이 틀리는 유형들의 문제를 중점적으로 공부해 틀리지 않도록 대비하는 것이 효과적인 방법이기 때문이다. 철수의 방법이 AI 모델 학습을 위한 일반적인 데이터 라벨링 방식이라면 영희의 학습 계획이 능동 학습이라고 말할 수 있다. 위의 예시와 같이 풀 수 있는 문제의 수가 제한된 것처럼 라벨링할 수 있는 데이터의 수가 제한된 상황에서는 성능 향상에 효과적인 데이터를 선별하는 과정이 중요하다.

능동 학습은 학습 데이터 중 모델 성능 향상에 효과적인 데이터들을 선별한 후 선별한 데이터를 활용해 학습을 진행하는 방법이다. 학습 데이터를 확보하는 과정은 데이터를 수집하는 것과 수집한 데이터에 라벨을 태깅하는 라벨링 작업으로 구성돼 있다. 일반적으로 라벨링 작업은 많은 시간과 인적 자원 활용 비용이 소요된다. 라벨링 작업에 특정 도메인의 전문성이 요구된다면 더더욱 많은 비용을 필요로 할 것이다. 그렇기에 같은 수의 데이터에 라벨을 붙여 학습할 때 성능이 높게 나올 수 있도록 데이터를 선별한다면 효과적으로 딥러닝 모델을 학습할 수 있다. 이렇게 효과적인 데이터를 선별하는 방법을 연구하는 것이 능동 학습에 대한 연구다. 이와 반대로 주어진 라벨 데이터만 가지고 모델을 학습하는 방법을 수동 학습(passive learning)이라고 한다.

〈표2〉 능동 학습의 동기와 목적

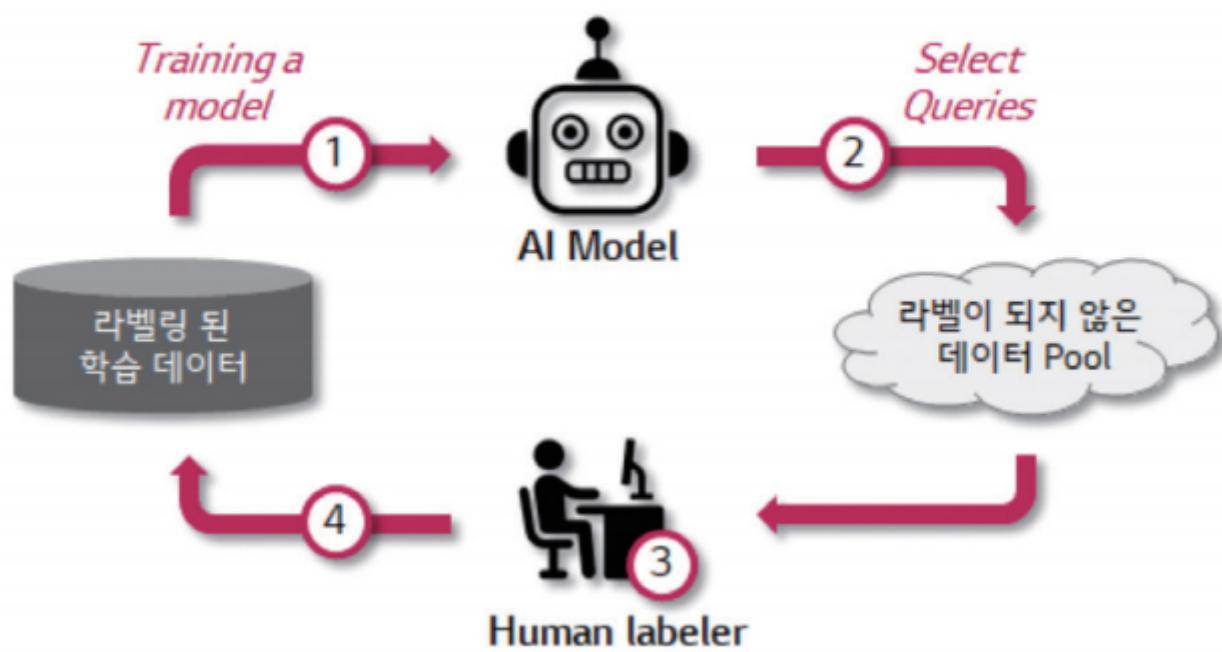
Motive

모델이 잘 맞히기 어려운 데이터를 찾아 학습한다면 더 적은 훈련 시간으로 더 좋은 성능을 낼 수 있을 것이다.

Objective

라벨링을 위한 예산이 한정됐을 때 모델의 성능을 극대화할 수 있는 라벨링 대상 데이터를 찾기

〈그림1〉 능동 학습의 4단계



능동 학습의 절차

능동 학습은 크게 4단계로 구성된다.

1. 모델 훈련하기 : 초기 학습 데이터(labeled data)를 이용해 모델을 학습한다.

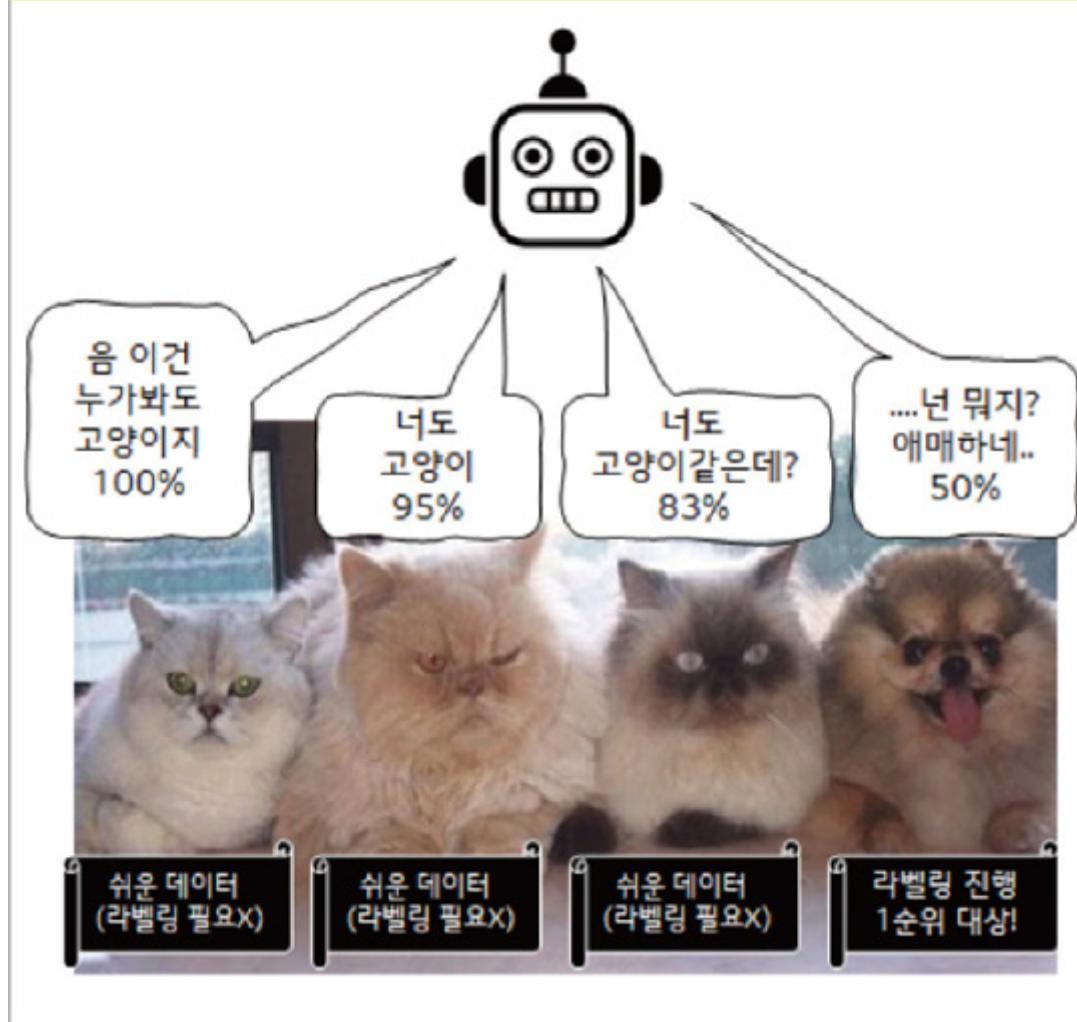
2. 쿼리 선택하기 : 라벨이 되지 않은 데이터 풀로부터 모델에게 도움이 되는 데이터를 선별한다.

3. 라벨링 : 선별한 데이터를 사람이 확인해 라벨을 태깅한다.

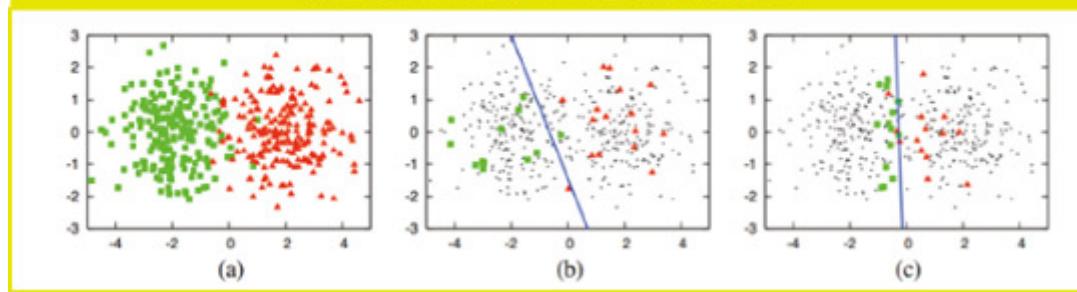
4. 선별한 라벨 데이터를 기존 학습 데이터와 병합한 후 다시 모델을 학습한다.

목표로 하는 성능이 나올 때까지 위의 방법을 반복해 수행한다.

〈그림3〉 모호한 데이터는 모델이 분류에 대한 확신이 낮을 것이다 (불확실한 추론)



〈그림2〉 모델 학습에 효과적인 데이터 선별하기



쿼리 전략(query strategy)

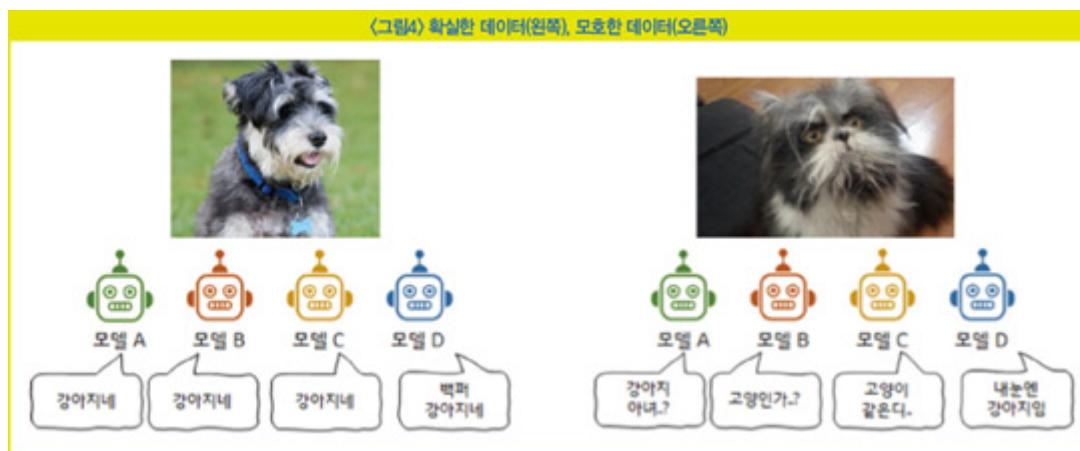
능동 학습의 핵심은 성능 향상에 효과적인 데이터를 선별하는 방법이다. 이러한 데이터 선별 방법을 '쿼리 전략(query strategy)'이라고 한다.

를 살펴보자. (a)는 2차원 평면에 나타낸 두 집단의 분포다. 여기에서 초록색 네모 집단과 붉은 세모 집단을 구별하는 모형을 만드는 것이 목표다. 이해를 돋기 위해 그림에는 집단이 표시돼 있지만 우리는 이러한 실제 모분포를 알 수 없다. 우리는 두 집단이 있다는 것만 알고 일부 데이터를 샘플링해 이 데이터가 초록 네모인지, 붉은 세모인지 라벨링한 뒤 해당 데이터만으로 두 집단을 구별하는 모델을 만들고자 한다.

(b)는 마구잡이로 데이터를 랜덤으로 추출해 초록 네모인지, 붉은 세모인지 라벨링한 것이다. 검은 점들은 선택되지 않은 데이터(unlabeled data)이기 때문에 어떤 집단인지 알 수 없다. 우리는 알고 있는 초록 네모와 붉은 세모의 샘플만으로 모델을 학습하게 된다. 그 결과 두 집단을 전반적으로 나눌 수 있는 선 하나를 (b)처럼 그려볼 수 있다. 하지만 (b)의 선은 실제 모집단의 분포를 70% 만 제대로 분류할 수 있다.

(c)도 마찬가지로 일부 데이터만 샘플링해 라벨링하고 분류 모델을 만들게 되는데 이 경우엔 샘플을 마구잡이로 고르는 것이 아니고 일정 기준에 따라 샘플링하게 된다. 이때의 기준은 '어떤 집단에 속하는지 모호하고 헷갈리는 데이터'인데 (c) 그림에서 볼 때 집단 간 경계에 있는 모호한 샘플 위주로 추출된 것을 볼 수 있다. 모델이 헷갈릴 만한 데이터 위주로 추출해 정답을 알려주고 학습한다면 더 정교한 분류 모델이 만들어질 수 있다. 그 결과 (c)의 모델은 모집단의 분포를 90% 제대로 분류할 수 있도록 학습됐다.

쿼리 전략을 어떻게 정하느냐에 따라 선별할 데이터가 달라진다. 대표적인 쿼리 전략에 대해 간단히 알아보자.



1. 불확실성에 기반한 샘플링

불확실성 샘플링(uncertainty sampling) 방식은 가장 단순한 쿼리 전략이다. AI 모델은 가장 불확실하다(least certain)고 생각하는 데이터를 추출해 라벨링이 필요하다고 요청한다. 예를 들어 강아지와 고양이를 분류하는 이진 분류(binary classification) 태스크에서는 모델이 어떤 데이터에 대해 강아지일 확률과 고양이일 확률을 각각 50% 내외로 추론한다면 해당 데이터는 강아지인지 고양이인지 모호한 데이터일 것이다. 이런 데이터를 라벨링해 모델에게 알려준

다면 분류 성능을 높이는 데 도움이 될 수 있다.

2. 의견 불일치를 고려한 샘플링

여러 AI 모델 간의 의견 불일치를 종합 고려해 라벨링 대상을 추출하는 방식 (query by committee)도 있다. 여러 모델 간 추론한 결과 불일치가 많은 데이터일수록 가장 헷갈리는 데이터, 즉 라벨링을 진행할 대상이 되는 것이다.

강아지와 고양이를 분류하는 모델을 여러 개 학습했다고 하자. 어떤 데이터를 넣었을 때 학습한 모델간 추론 의견이 일치한다면 그 데이터는 확실히 강아지 이거나 고양이인 데이터일 것이다. 이런 데이터는 라벨링을 진행하지 않아도 이미 여러 모델이 잘 맞춘다는 뜻이므로 넘어가도 좋다. 하지만 어떤 데이터를 넣었을 때 모델 간 추론 결과가 제각각이라면 그 데이터는 고양이인지 강아지인지 모호한, 정보가 많은 데이터다. 이때 라벨링을 진행해 모델 학습에 이용하면 분류 성능 향상에 도움이 될 수 있다.

이 밖에 다양한 쿼리 전략이 있지만 어떤 방법이든 간에 정보가 가장 많은 데이터를 선정해 라벨링해야 모델 학습에 도움이 될 것이라는 생각은 동일하다.

데이터 자체는 손쉽게 대량으로 확보할 수 있지만 모델이 학습에 사용할 수 있는 '유의미한 라벨 정보가 포함된 데이터'는 극소수다. 데이터는 많지만 역설적이게도 AI 모델 학습을 위해 '쓸모 있는 데이터'는 많지 않다. 풀고자 하는 태스크를 위한 라벨 정보를 새로 만드는 것은 시간과 비용에 의해 현실적으로 불가능한 것이 대부분이다. 이러한 어려움을 조금이라도 해결하기 위해 연구해 온 방법이 능동 학습이다.

능동 학습은 그 자체로 어떤 딥러닝 기술이라기보다 효과적인 학습을 위한 시스템이라고 보는 것이 맞다. 그리고 그 시스템의 일부분에서 반드시 인간의 라벨링 작업을 필요로 한다. AI는 어떤 식으로든 이와 같이 조금이라도 사람의 도움(라벨링과 같은)을 필요로 한다. 여러 사례들을 보면 능동 학습은 AI 모델 학습을 시작하는 초기 개발 단계에 매우 효과적이다. 어차피 가르쳐 줘야 할 것이라면 조금이라도 효율적으로, 더 도움이 될 수 있는 방향으로 작업을 할 수

있는 것이 좋지 않을까.

능동 학습 방법만으로 데이터에 관한 모든 문제를 해결할 수는 없겠지만 AI 기술이 효율적으로 접목될 새로운 가능성을 열어 줄 수 있다.

[본 기사는 한경비즈니스 제 1308호(2020.12.21 ~ 2020.12.27) 기사입니다.]

© 매거진한경, 무단전재 및 재배포 금지

#AI테크

[김민주 기자](#)