

[그림 2. 골고루 배워놓는 사전학습(Pre-training)]

## 8.2 대규모 데이터에 대한 Pre-training

어떤 후속 태스크에 적용하는 잘 수행할 수 있도록 하려면 방대한 양의 지식을 골고루 배워놓는 것이 좋습니다.

따라서 모델의 사전학습은 대규모의 오픈도메인 데이터에 대해 이뤄지는 것이 일반적인데요,

이미지와 텍스트에 대한 대표적인 사전학습 데이터와 태스크는 다음과 같습니다.

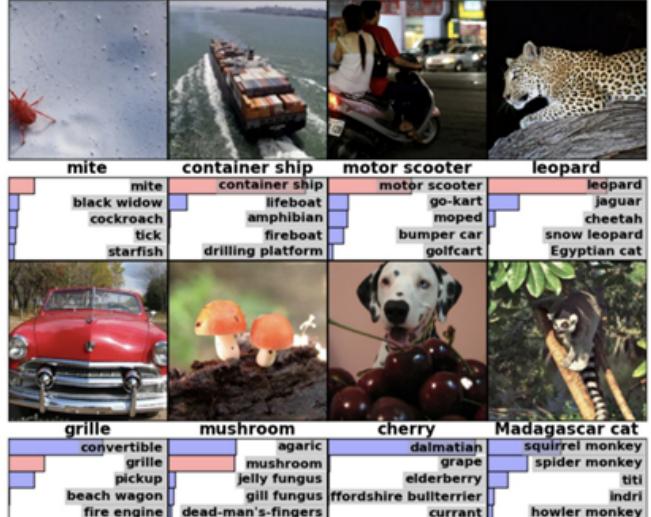
### 8.2.1 시각 데이터에 대한 사전학습

테크레터에서도 벌써 여러 번 등장한 **이미지넷**([참고\(see page 0\)](#)) 데이터 인식 대회는 가장 유명한 이미지 사전학습 과제라고 할 수 있습니다.

120만 장의 학습용 이미지를 가지고 1000개 카테고리로 분류하는 이 과제는 대표적인 인공지능 이미지 인식 태스크입니다.



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



<b>mite</b>	<b>container ship</b>	<b>motor scooter</b>	<b>leopard</b>
black widow cockroach tick starfish	lifeboat amphibian fireboat drilling platform	motor scooter go-kart moped bumper car golfcart	leopard jaguar cheetah snow leopard Egyptian cat
<b>grille</b>	<b>mushroom</b>	<b>cherry</b>	<b>Madagascar cat</b>
convertible grille pickup beach wagon fire engine	agaric mushroom jelly fungus gill fungus dead-man's-fingers	dalmatian grape elderberry ffordshire bulterrier currant	squirrel monkey spider monkey titi indri howler monkey

**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

[그림 3. ImageNet dataset 분류 대회(ILSVRC<sup>46</sup>)]

120만 장이나 되는 컬러 이미지를 1000개나 되는 카테고리로 분류하도록 학습된 모델은 일반적인 이미지의 특징 대부분을 다뤄봤다고 봐도 무방합니다.

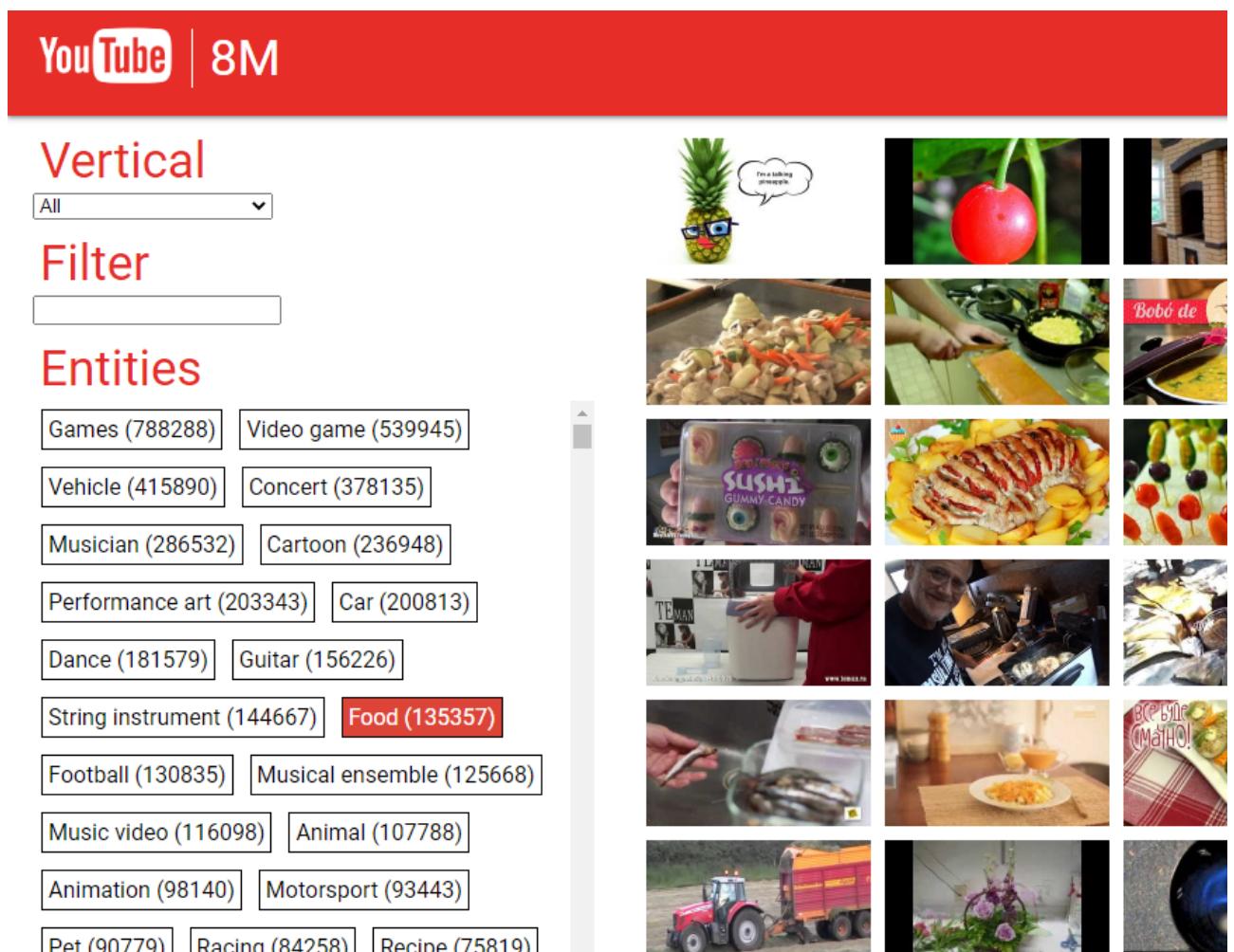
어떤 색상들이 나타나는지, 등장하는 사물의 직선, 곡선, 또한 이들이 합쳐져서 이루는 도형, 큰 개체와 작은 개체, 전경과 배경 등을 학습했겠지요.

다양한 내용을 두루 살펴보았으니 이미지를 대상으로 하는 어떤 태스크에 적용된다고 해도 사전지식을 기반으로 빠르게 학습할 수 있을 것입니다.

정지된 이미지 뿐 아니라 동영상에 대해서도 비슷한 사전학습용 데이터가 있습니다.

구글이 공개한 **Youtube-8M**은 무려 총 35만 시간에 달하는 610만개의 비디오를 약 3800개의 카테고리로 다중 분류하는데 활용할 수 있습니다.

46 <http://www.image-net.org/challenges/LSVRC/>



[그림 4. YouTube-8M dataset ([링크](#)<sup>47</sup>)]

### 8.2.2 언어 데이터에 대한 사전학습

언어에 대해 전반적으로 배워놓는다는 것은 문맥에 따라 활용되는 단어의 의미, 뉘앙스, 적절한 문체 등을 습득한다는 것입니다.

자연어의 경우 이미지나 비디오 데이터와는 달리, 언어권의 차이로 인해 데이터를 언어권별로 각각 수집하여 학습시켜야 하는 문제가 있습니다.

영어의 경우 공개된 데이터가 많지만 한국어나 기타 언어에 대해서는 다량의 표준 데이터를 구하기 쉽지 않죠.

하지만 무난하게 활용하기 좋은 데이터로는 **위키피디아(Wikipedia)**가 있습니다.

47 <https://research.google.com/youtube8m/index.html>

Project page [Talk](#) [Read](#) [View source](#) [View history](#) [Search Wikipedia](#)

# Wikipedia:Database download

From Wikipedia, the free encyclopedia

*For scheduling, related tools etc., see [m:Data dumps](#).*

**This help page is a how-to guide.**  
It details processes or procedures of some aspect(s) of Wikipedia's norms and practices. It is not one of Wikipedia's policies or guidelines, and may reflect varying levels of consensus and vetting.

Shortcuts  
[WP:DUMP](#)  
[WP:DUMPS](#)

Wikipedia offers free copies of all available content to interested users. These databases can be used for mirroring, personal use, informal backups, offline use or database queries (such as for [Wikipedia:Maintenance](#)). All text content is multi-licensed under the [Creative Commons Attribution-ShareAlike 3.0 License \(CC-BY-SA\)](#) and the [GNU Free Documentation License \(GFDL\)](#).

**Readers' FAQ**

About Wikipedia (Administration · FAQs) · Authority control · Books · Categories · Censorship · Copyright · Disambiguation · Images and multimedia · ISBN ·

[그림 5. 위키피디아 덤프 데이터셋 다운로드([링크<sup>48</sup>](#))]

위키피디아의 경우 전 분야에 걸친 백과사전 지식을 대상으로 하기 때문에 텍스트 모델 사전학습을 시킬 수 있는 대표적인 데이터입니다.

한국어로도 다운받을 수 있고, 이외 여러 언어에 대해서도 제공하고 있습니다.

한국어의 경우 **나무위키** 데이터도 다운로드 받을 수 있는데, 조금 더 구어체에 가깝고 자연스러운 표현을 학습할 수 있습니다.

이외에 국립국어원에서 공개하는 **세종말뭉치**가 있습니다.

48 [https://en.wikipedia.org/wiki/Wikipedia:Database\\_download](https://en.wikipedia.org/wiki/Wikipedia:Database_download)

말뭉치 분류	매체 및 장르(1단계)	매체 부가정보(2단계)
현대문어	잡지(주간지, 월간지, 계간지)	경제, 과학, 국제, 기타(독자투고, 인물, 화제, 응대 등), 문화, 보도, 사설, 사회, 생활, 스포츠, 연극, 오피니언, 정치, 총류, 취미, 칼럼
현대문어	잡지(주간지, 월간지, 계간지)	교육자료, 사회, 상상적 텍스트, 생활, 예술론, 인문, 자연, 체험기술적 텍스트, 총류
현대문어	책	교육자료, 기타(독자투고, 인물, 화제, 응대 등), 사회, 상상적 텍스트, 생활, 수필, 예술, 예술론, 인문, 자연, 정보, 체험 기술적 텍스트, 총류
현대문어	기타 출판물 (안내문, 소책자, 정부 문서 등)	예술론, 인문, 실기 르포
현대문어	화면이 있는 방송 녹화 전사	생활 대화, 사회 대화, 녹화/사회, 총류
현대구어	화면이 있는 방송 녹화 전사	사회, 생활, 예술론, 인문, 총류
현대문어	화면이 없는 방송 녹음 전사	예술론, 총류
현대구어	화면이 없는 방송 녹음 전사	사회, 생활, 신문, 예술론, 인문, 자연, 체험기술적 텍스트, 총류
현대문어	기타 녹음 전사	예술론
현대구어	기타 녹음 전사	-
현대문어	전자출판물	과학, 사회, 생활, 예술론, 인문, 자연, 체험기술적 텍스트, 토론, 총류

[그림 6. 국립국어원 세종말뭉치([링크](#)<sup>49</sup>)]

여러 매체로부터 모은 현대 문어/구어체를 제공하고 있어서 경우에 따라 활용하기 좋습니다.

이외에 뉴스, 리뷰 등의 데이터도 자주 활용되는 사전학습용 데이터입니다.

### 8.3 Self-Supervised Learning: 나 혼자 어떻게든 해볼게

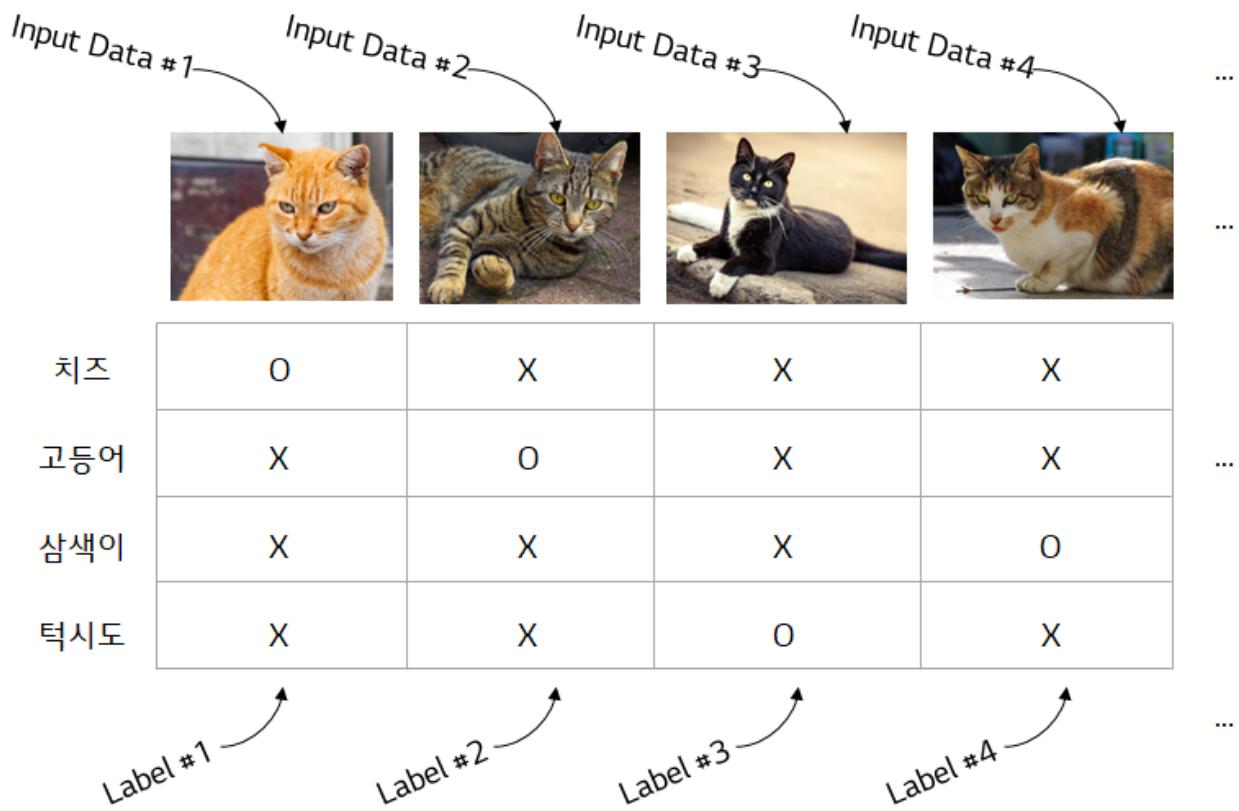
하지만 대규모로 구할 수 있는 데이터라 해도 라벨까지 잘 달려있는 경우는 드뭅니다.

라벨(label)이라 함은 데이터에 대해 인공지능이 예측하기를 희망하는 결과입니다.

예를 들어, 한국 길고양이 종류를 구별한다고 가정하겠습니다.

학습 목적은 고양이 사진을 넣었을 때 어떤 종류의 고양이인지 추론하는 것입니다.

<sup>49</sup> <https://ithub.korean.go.kr/user/guide/corpus/guide1.do>



[그림 7. 길고양이 종류 구별용 데이터에 대한 Input data와 Label]

[그림 7]에서 보이는 사진은 인공지능이 인식할 대상으로, 입력 데이터(Input Data)라고 합니다.

라벨(Label)은 인공지능이 입력 데이터를 제공받으면 추론해야 할 결과로, 분류 과제의 경우 각 카테고리당 0/X 또는 1/0으로 구별합니다.

고양이 종류 구분 과제의 경우 라벨을 길이 4만큼의 벡터로 만들 수 있습니다.

입력 데이터만으로 학습할 수 있는 모델의 종류는 많지 않기 때문에, 정답 라벨이 제대로 달려 있는 데이터를 얼마나 모을 수 있느냐가 모델 학습의 품질을 좌우합니다.

하지만 정답 라벨링은 사람이 일일히 만들어줘야 해서 공수가 많이 드는 작업이지요.

어려운 일은 아니지만 인형의 눈알을 하나씩 붙이고, 피자박스를 접고, 마늘 껍질을 하나씩 깨고 다듬는 것처럼 귀찮고 시간이 많이 드는 단순반복 노동입니다.



[그림 8. 중국에서는 마늘까기를 교도소 수감자에게 시킨다고 합니다. 사소하지만 귀찮고 양 많은, 누군가는 해야하는 일 이지요... (자료: 넷플릭스-Rotten<sup>50</sup>)]

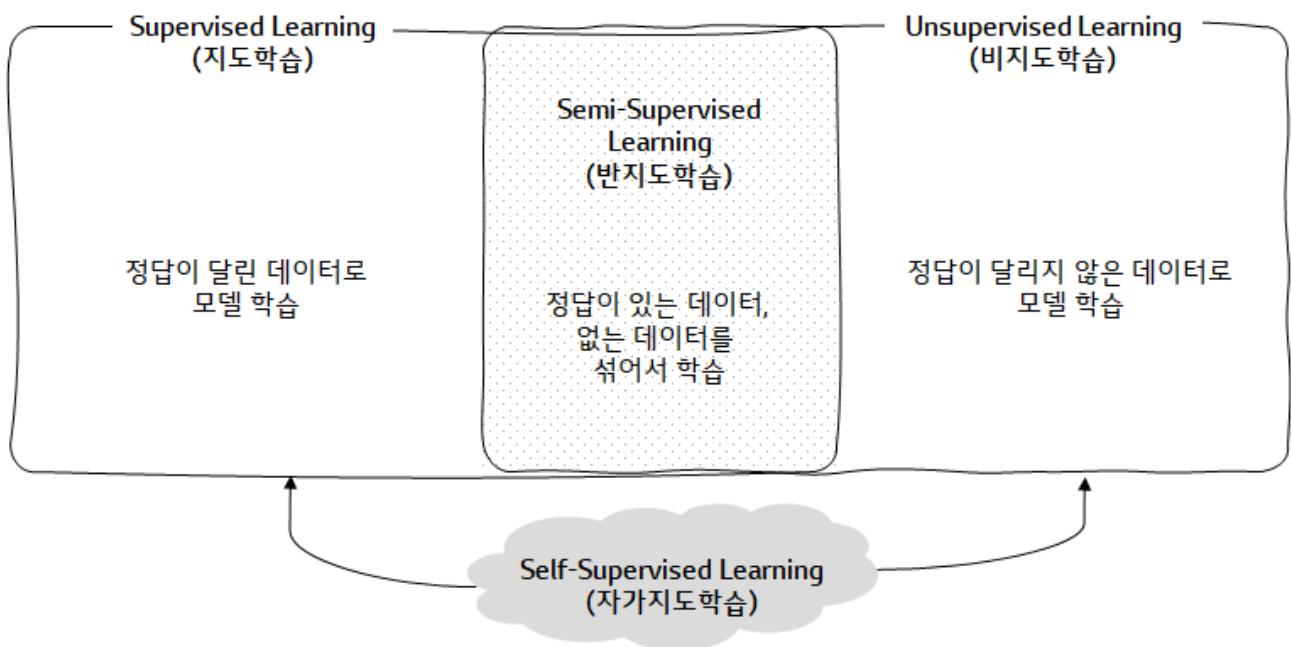
라벨을 붙인 데이터를 많이 만들기는 힘들어도 라벨 없는 데이터를 모으는 것은 어렵지 않습니다.

수많은 이미지와 동영상, 텍스트 문서 자체는 하루에도 셀 수 없는 양이 쏟아지고 있으니까요.

일단 데이터가 많기는 한데 인공지능 모델에게 알려줄 정답은 없고, 어떻게든 뭔가 활용할 방법이 없을까요?

이 때 활용할 수 있는 학습 방법이 **Self-Supervised Learning(자가지도학습)**입니다.

50 [https://en.wikipedia.org/wiki/Rotten\\_\(TV\\_series\)](https://en.wikipedia.org/wiki/Rotten_(TV_series))



[그림 9. 인공지능 모델 학습 방법의 종류]

Self-Supervised Learning이라는 용어를 처음 듣는다면 마치 인공지능이 스스로 학습하여 똑똑해지는 것처럼 느껴집니다만, 그런 거창한 개념은 아닙니다.

Self-Supervised Learning은 사람이 만들어주는 정답 라벨이 없어도 기계가 시스템적으로 자체 라벨을 만들어서 사용하는 학습 방법입니다.

사람이 라벨을 만들어줄 필요가 없다는 점에서는 Unsupervised Learning으로 볼 수 있지만, 자체적으로 라벨을 만들어 사용한다는 점에서 Supervised Learning의 일종으로 볼 수도 있습니다.

예를 들어 보면 더 잘 이해할 수 있습니다. 다음은 Self-Supervised Learning의 경우입니다.

### 8.3.1 예: 이미지 데이터를 위한 Self-Supervised Learning

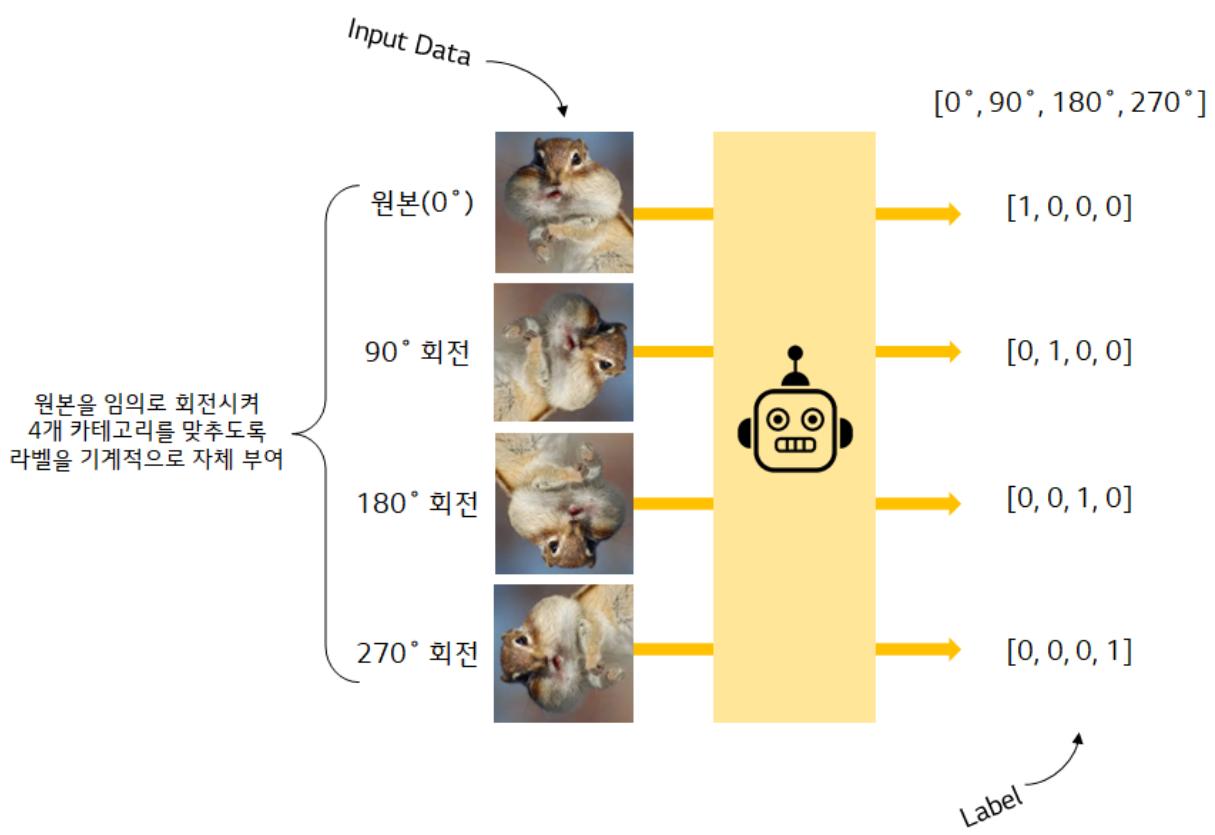


vs

다람쥐와 청설모를 구별하고 싶은데, 우선 종류에 상관 없이 설치류 짐승의 사진을 10만 장 정도 충분히 많이 확보해 두었다.

하지만 10만 장의 설치류 사진이 각각 어떤 종류에 해당하는지 라벨을 부여하기는 시간과 비용을 확보하기 어려웠다. 데이터는 많고... 설치류의 일반적인 특징에 대해 뭐라도 학습한 딥러닝 모델을 사전학습하려고 한다.

이 때 **Self-Supervised Learning**을 활용하여 자동으로 라벨을 부여하고 맞출 수 있는 태스크를 만들어 모델을 사전 학습시키기로 하였다.

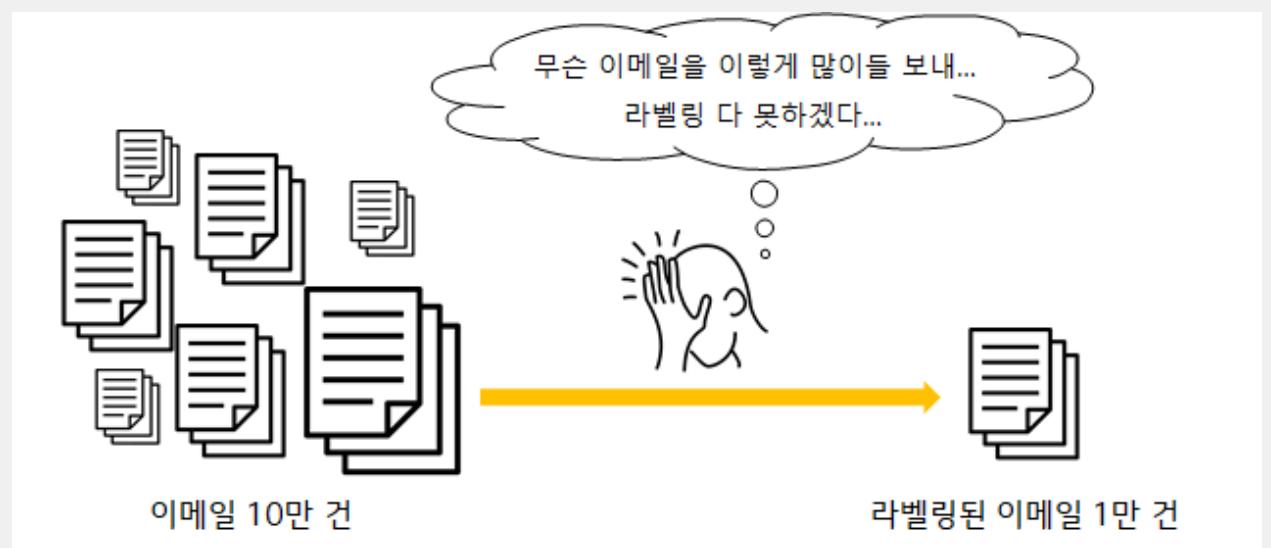


이렇게 사전학습한 모델을 다림쥐/청설모 데이터로 Transfer Learning 하였더니 다림쥐/청설모만으로 학습한 모델 보다 좋은 성능을 얻을 수 있었다.

### 8.3.2 예: 텍스트 데이터를 위한 Self-Supervised Learning

사외로 전송되는 이메일의 보안 위반 여부를 검출하고자 하는데, 우선 보안 위반 여부와 관계없이 사외전송 이메일 10만 건을 모아두었다.

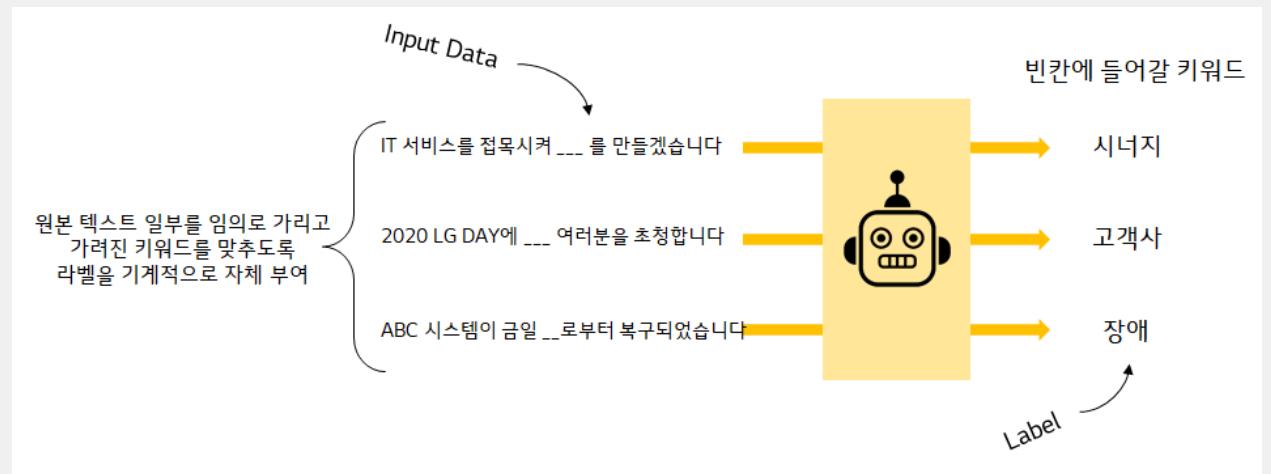
하지만 10만 건의 이메일을 전부 살펴보기 힘들어 1만 건의 이메일에 대해서만 라벨링을 할 수 있었다.



가진 데이터를 전부 활용하여 조금이라도 업무 관련 도메인 키워드를 학습할 수 있도록 Self-Supervised Learning으로 사전학습을 시키고자 한다.

사전학습 태스크로는 메일의 중간 단어를 빈칸으로 대체한 후 들어갈 단어를 알아맞추도록 하였다.

이렇게 하니 10만건의 메일을 전부 활용하여 인공지능 모델에게 우리 회사에서 자주 쓰는 키워드를 인식시킬 수 있었다.



이렇게 만든 모델을 Transfer Learning으로 활용하니 1만 건의 라벨링 데이터만으로 학습한 모델보다 좋은 성능을 보였다.

이처럼 Self-Supervised Learning은 주로 사전학습에서 이용되며 다양한 데이터는 있으나 라벨은 없는 경우에 활용할 수 있습니다.

Self-Supervised Learning의 과제 자체가 의미 있는 것은 아니나, 수많은 데이터를 자체 라벨링으로 학습하게 되면 해당 데이터에 대한 전반적인 지식을 넓고 얕게 습득할 수 있게 되는 것이지요.

이렇게 학습한 모델을 향후 후속 과제로 Transfer Learning하면 맨 땅에서 학습한 모델에 비해 일반적으로 좋은 성능을 보입니다.

### 8.3.3 예: Google BERT(Bidirectional Encoder Representations from Transformers)

Self-Supervised Learning 기법으로 사전학습을 하고 다양한 태스크에 Transfer Learning을 할 수 있는 대표적인 예로 Google의 '**BERT**'라는 모델이 있습니다.

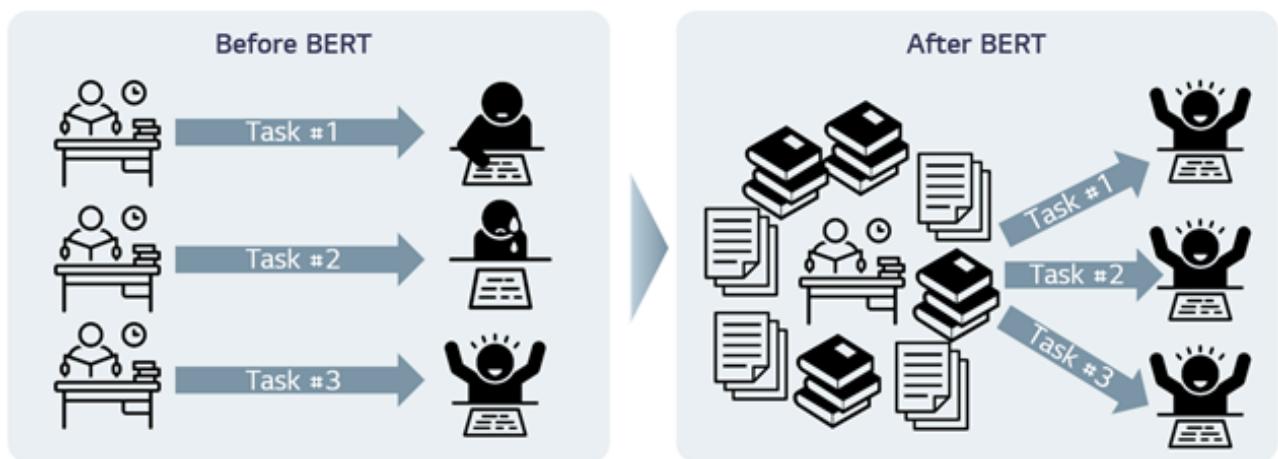
BERT는 자연어처리 연구 패러다임을 전환한 계기가 된 모델입니다. 인공지능에서의 자연어 처리는 BERT 이전과 이후로 나뉘죠.

예전부터 사전학습과 Transfer Learning의 개념이 있긴 했으나, 기존의 사전학습이 워드 임베딩 등 그저 보조적 역할을 수행하는 느낌이었다면 BERT는 사전학습 자체가 주가 되는 모델입니다.

기존 모델 학습 방식을 시험에 비유해보겠습니다.

고3은 수능을 위해 수능 기출을 따로 풀고, 토익 응시생은 토익 문제만 엄청 풀고, LGCNS 사원은 정보처리기사를 공부해서 각각의 시험에서 성적을 내기 위해 노력했습니다.

하지만 BERT의 관점은 '이거저거 닥치는대로 책을 많이 본 사람'이 나중에 '어떤 시험을 쳐도 잘 보게 된다'는 것입니다.



[그림 10. (왼)BERT 이전의 자연어처리 모델 학습방식, (오)BERT의 학습 방식]

즉 '언어'라는 분야 전반에 걸쳐 지식을 두루 쌓은 '하나의 거대한 뇌'를 사전학습으로 만든다는 개념입니다.

BERT는 사전학습에서 상당한 양의 데이터(텍스트 코퍼스)를 커다란 모델로 학습시켰으며, 후속 태스크를 위한 Transfer Learning은 간략하게만 진행해도 좋은 성능을 낼 수 있었습니다.

무려 11개의 자연어처리 과제에서 1위를 했는데, 이는 텍스트를 대상으로 할 수 있는 거의 대부분의 과제라고 볼 수 있습니다.

이 때 BERT가 사전학습한 문서가 무려 33억 단어만큼이며, 16개의 TPUv3 칩을 활용하여 학습하였다고 합니다.

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	Average -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
BERT <sub>BASE</sub>	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
BERT <sub>LARGE</sub>	<b>86.7/85.9</b>	<b>72.1</b>	<b>91.1</b>	<b>94.9</b>	<b>60.5</b>	<b>86.5</b>	<b>89.3</b>	<b>70.1</b>	<b>81.9</b>

[그림 11. BERT 모델 공개 당시 모든 과제에서 성능 1위를 기록하였습니다.]

어찌 생각해보면 당연한 결과로 보입니다.

전 과목을 다 못하는 꼴지인데, 수학 한 과목에서만 천재인 학생이 있을까요?

모두의 학창시절이 비슷했겠지만 모든 과목을 포기하지 않고 두루 공부하는 학생이 각 과목의 상위권 또한 차지하곤 합니다.

지식이라는 것은 서로 연결되는 부분이 있어서 한 부분에서 습득했던 내용이 전혀 예기치 못한 다른 영역을 배우는 데 도움을 줄 수 있기 때문이지요.

일반적으로 Self-Supervised Learning을 활용한 Pre-Trained 모델은 다양한 방대한 지식을 골고루 습득하는 것을 목적으로 하기에

대체로 모델의 사이즈가 큰 편이고 사전학습 규모가 어마어마하다는 특징이 있습니다.

어떻게 보면 GPU 학습 장비나 데이터 저장공간에 대한 비용 부담이 커서 실용적이지 않게 느껴지기도 합니다.

하지만 사전학습 모델은 한 번 잘 마련해놓으면 향후 어떤 과제든 적용할 수 있습니다.

장기적으로 볼 때 두고두고 여러군데 활용 가능한 Base 모델을 준비한다는 개념으로 생각해야 합니다.

## 8.4 마무리

딥러닝 기반의 AI 모델은 다양한 양질 데이터를 필요로 합니다.

또한 대부분은 모델의 학습을 위해 사람이 태깅한 정답 라벨을 필요로 합니다. (Human-labeled data)

데이터 자체를 많이 확보하기는 쉬울 지 몰라도, 라벨링 잘 된 데이터를 다양으로 구하는 것은 쉬운 일이 아닙니다.

이 때 활용할 수 있는 방법이 시스템적으로 라벨을 보유하고 학습할 수 있는 자가지도학습, Self-Supervised Learning입니다.

이 방식으로 모델은 전반적인 지식을 골고루 사전학습(Pre-Training)할 수 있으며, 향후 특정 태스크로 Transfer Learning 할 경우 좋은 성능을 보일 수 있습니다.

이번 시간은 Pre-Training과 Self-Supervised Learning에 대해 설명드렸습니다.

다음 시간에는 사람의 라벨링 작업을 최대한 효율적으로 할 수 있는 방법, '**Active Learning**'에 대해 자세히 알아보겠습니다.

감사합니다 😊

---

## 참고자료

- ImageNet Large Scale Visual Recognition Challenge (ILSVRC), <http://www.image-net.org/challenges/LSVRC/>
- Google Youtube-8M Dataset, <https://research.google.com/youtube8m/index.html>
- WikiPedia database dump, [https://en.wikipedia.org/wiki/Wikipedia:Database\\_download](https://en.wikipedia.org/wiki/Wikipedia:Database_download)
- 나무위키 데이터베이스 덤프, <https://namu.wiki/w/%EB%82%98%EB%AC%B4%EC%9C%84%ED%82%A4:%EB%8D%B0%EC%9D%B4%ED%84%B0%EB%B2%A0%EC%9D%B4%EC%8A%A4%20%EB%8D%A4%ED%94%84>
- 국립국어원 언어정보나눔터, <https://ithub.korean.go.kr/user/guide/corpus/guide1.do>
- What is the difference between self-supervised and unsupervised learning?, Quora, <https://www.quora.com/What-is-the-difference-between-self-supervised-and-unsupervised-learning>
- Conversational AI, Insight+, 2019, 김명지, [Conversational AI<sup>51</sup>](#)
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, J. Devlin, et al., 2018, <https://arxiv.org/abs/1810.04805>

---

<sup>51</sup> <https://wire.lgcns.com/confluence/display/TS12032247/Conversational+AI>

## 9 [9편] 족집게 데이터로 인공지능 학습하기

---

안녕하세요, CTO AI빅데이터연구소입니다.

한 달에 두 번씩 **AI 테크레터**를 통해 인공지능 지식을 임직원 여러분들께 공유드리고 있습니다.

모든 CNSer가 이해하실 수 있도록 쉽게 작성하려고 하니, 상세 기술에 대한 궁금증이 생기시면 댓글이나 이메일을 통해 언제든 연락 바랍니다 😊

본 업로드는 [TECH wiki AI개시판](#)(see page 7)에서 연재됩니다.

작성 : CTO AI빅데이터연구소 AI기술팀 [김명지 팀장/총괄 CONSULTANT](#)/언어AI LAB<sup>52</sup>

---

- [데이터의 바다, 정보의 홍수](#)(see page 135)
  - [Active Learning: 족집게 데이터로 공부하기](#)(see page 137)
    - [Active Learning의 절차](#)(see page 138)
    - [Query Strategy: 이 데이터를 제게 가르쳐 주십시오!](#)(see page 139)
      - [Uncertainty Sampling](#)(see page 140)
      - [Query by committee](#)(see page 141)
  - [마무리](#)(see page 142)
- 

오늘은 Human Labeling 공수를 최대한 효율적으로 하는 방법, Active Learning에 대해 알아보겠습니다.

지난 시간까지의 내용이 궁금하신 분은 ★[AI Tech Letter](#)(see page 7)★를 확인하시기 바랍니다.

### 9.1 데이터의 바다, 정보의 홍수

지난 시간에는 AI 스스로 라벨을 만들고 사전 학습할 수 있는 Self-supervised Learning에 대해 설명드렸습니다.

데이터 자체는 많지만, 인공지능 모델을 학습시킬 수 있을 만큼 정제되고 라벨이 잘 달려있는 데이터를 구하기는 어렵다는 데에서 착안된 기법인데요,

오늘 설명드릴 Active Learning(능동 학습)이라는 기술도 데이터는 많으나 인공지능을 '학습시킬 데이터'를 마련하기 쉽지 않을 때 이용할 수 있는 기술입니다.

---

<sup>52</sup><https://wire.lgcns.com/confluence/display/~78628>



[그림 1. 양이 많다고 다 좋은 데이터는 아닙니다]

강아지 이미지와 고양이 이미지를 구별하는 태스크라면 누구나 데이터를 라벨링 할 수 있어 적은 비용으로도 금방 데이터를 모으겠지만,

뇌 MRI 영상으로부터 파킨슨 병 여부를 판단하는 태스크라면 해당 분야의 전문 의사가 아니라면 몹시 어려운 일이지요. 게다가 이미지를 한장씩 보면서 파킨슨 병 여부를 일일이 태깅할만큼 시간 여유가 있는 뇌 전문의를 구한다는 것 또한 쉬운 일은 아닙니다.

하지만 뇌 전문의 다섯 명이 한 달 동안 매일 20장씩만 이미지를 보고 라벨링을 해줄 수 있다면 어떨까요?

수많은 뇌 MRI 영상 중 가능한 한 파킨슨 병인지 아닌지 가려내는데 효과적인 데이터만 뽑아서 정답을 알려달라고 하고 싶지 않나요?

**Active Learning**은 라벨링을 할 수 있는 인적 자원은 있지만, 많은 수의 라벨링을 수행할 수 없을 때 효과적으로 라벨링을 하기 위한 기법입니다.

수행하고자 하는 태스크가 너무나 특수하여 해당 도메인의 전문 인력만이 데이터를 라벨링 할 수밖에 없는 경우, 최대한 학습에 효과적인 데이터만을 뽑아내는 데에 쓰일 수 있습니다.

뇌 MRI 영상으로부터 파킨슨 병이라고 판단할 수 있을만한 대표적인 특징이 있겠지요.

하지만 누구나 파악할 수 있는 대표적인 특징을 가진 데이터 말고, 너무 애매모호해서 전문의가 아니고서는 판단하기 어려운 데이터를 확보할 수 있다면 인공지능 학습에 도움이 되지 않을까요?

## 9.2 Active Learning: 족집게 데이터로 공부하기

수능일이 얼마 남지 않았습니다.

수능일까지 남은 시간 동안 각 수험생이 풀어볼 수 있는 문제의 수는 한정되어있습니다.

고3 수험생 철수와 영희가 있다고 생각해봅시다.

철수와 영희는 각각 동일한 문제집 10권을 가지고 있습니다. 문제집 1권 당 문제가 100개씩 있습니다.

철수와 영희는 이 중 총 500 문제를 풀고 난 후 수능을 보게 됩니다.

철수의 학습 계획	철수는 문제집 5권을 임의로 뽑아서 안에 있는 문제를 전부 풀었습니다.
영희의 학습 계획	<p>영희는 문제집 3권을 임의로 뽑아서 안에 있는 문제를 전부 풀었습니다.</p> <p>영희는 풀었던 문제집 3권에서 많이 틀리는 유형들을 체크하였습니다.</p> <p>그 뒤 남은 7권의 문제집에서 많이 틀리는 유형에 해당하는 200문제를 추가로 찾아 풀었습니다.</p>

철수와 영희의 학습 방법이 위와 같을 때, 누가 더 효과적으로 학습했을까요?

일반적으로 본다면 영희가 더 효과적으로 학습했다고 말할 수 있습니다.

풀 수 있는 문제가 500개로 제한된 상황이라면 많이 틀리는 유형들의 문제를 중점적으로 공부해 틀리지 않도록 대비하는 것이 효과적인 방법이기 때문입니다.

철수의 방법이 AI 모델 학습을 위한 일반적인 데이터 라벨링 방식이라면 영희의 학습 계획이 Active Learning이라고 말할 수 있습니다.

위의 예시와 같이 풀 수 있는 문제의 수가 제한된 것처럼, 라벨링 할 수 있는 데이터의 수가 제한된 상황에서는 성능 향상에 효과적인 데이터를 선별하는 과정이 중요합니다.

**Motive**

모델이 잘 맞추기 어려운 데이터를 찾아 학습한다면,  
더 적은 훈련시간으로 더 좋은 성능을 낼 수 있을 것이다.

**Objective**

Labeling을 위한 예산이 한정되었을 때,  
모델의 성능을 극대화할 수 있는 Labeling 대상 데이터를 찾기

[그림 2. Active Learning의 동기와 목적]

Active Learning은 학습 데이터 중 모델 성능 향상에 효과적인 데이터들을 선별한 후, 선별한 데이터를 활용해 학습을 진행하는 방법입니다.

학습 데이터를 확보하는 과정은 데이터를 수집하는 것과 수집한 데이터에 라벨을 태깅하는 라벨링 작업으로 구성되어 있습니다.

일반적으로 라벨링 작업에 많은 시간과 인적 자원 활용 비용이 소요됩니다. 라벨링 작업에 특정 도메인의 전문성이 요구된다면 더더욱 많은 비용을 필요로 하겠지요.

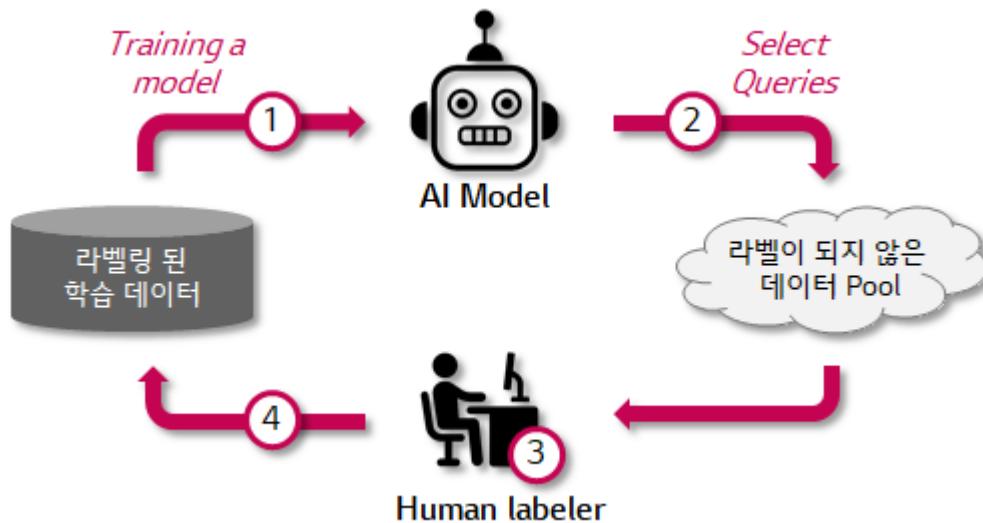
그렇기에 같은 수의 데이터에 라벨을 붙여서 학습할 때, 성능이 높게 나올 수 있도록 데이터를 선별한다면 효과적으로 딥러닝 모델을 학습할 수 있습니다.

이렇게 효과적인 데이터를 선별하는 방법을 연구하는 것이 Active Learning입니다.

이와 반대로 주어진 라벨 데이터만 가지고 모델을 학습하는 방법을 Passive Learning(수동 학습)이라고 합니다.

### 9.2.1 Active Learning의 절차

Active Learning은 크게 4단계로 구성되어 있습니다.



[그림 3. Active Learning의 4단계]

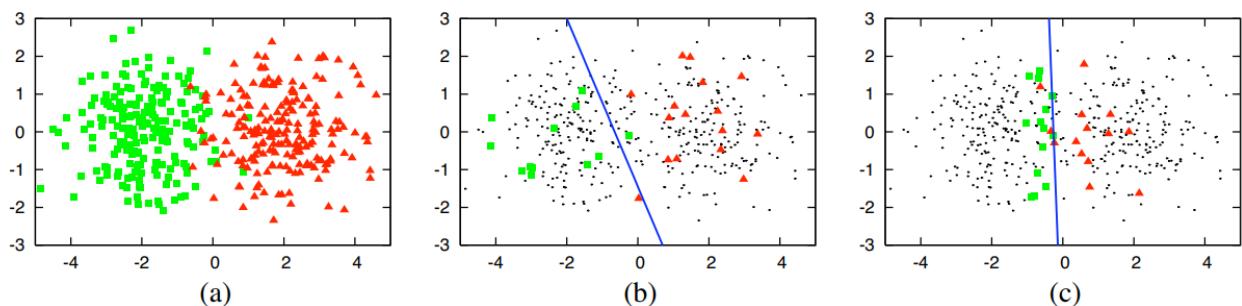
1. Training a Model : 초기 학습 데이터(labeled data)를 이용해 모델을 학습합니다.
2. Select Query : 라벨이 되지 않은 데이터 폴로부터 모델에게 도움이 되는 데이터를 선별합니다.
3. Human Labeling : 선별한 데이터를 사람이 확인하여 라벨을 태깅합니다.
4. 선별한 라벨 데이터를 기준 학습 데이터와 병합한 후, 다시 모델을 학습합니다.

목표하는 성능이 나올 때까지 위의 방법을 반복해 수행합니다.

### 9.2.2 Query Strategy: 이 데이터를 제게 가르쳐 주십시오!

Active Learning의 핵심은 성능 향상에 효과적인 데이터를 선별하는 방법입니다.

이러한 데이터 선별 방법을 ‘쿼리 전략(Query Strategy)’이라고 합니다.

[그림 4. 모델 학습에 효과적인 데이터 선별하기. 자료<sup>53]</sup>

53 <http://burrsettles.com/pub/settles.activelearning.pdf>

위 [그림 4]를 예로 들어보겠습니다. (a)는 2차원 평면에 나타낸 두 집단의 분포입니다.

우리는 초록색 네모 집단과, 붉은 세모 집단을 구별하는 모형을 만들고자 합니다. 하지만 우리는 이러한 실제 모분포를 알 수 없습니다. (이해를 돋기 위해 그림에는 표시했지만요!)

우리는 두 집단이 있다는 것만 알고, 일부 데이터를 샘플링해와서 이 데이터가 초록 네모인지, 붉은 세모인지 라벨링하고 해당 데이터만으로 두 집단을 구별하는 모델을 만들고자 합니다.

(b)는 마구잡이로 데이터를 랜덤하게 샘플링해와서 초록 네모인지 붉은 세모인지 라벨링한 것입니다. 검은 점들은 선택되지 않은, Unlabeled data이기 때문에 어떤 집단인지 알 수 없습니다.

우리는 알고 있는 초록 네모와 붉은 세모 샘플만으로 모델을 학습하게 됩니다. 그 결과 두 집단을 전반적으로 나눌 수 있는 선 하나를 (b)처럼 그리게 됩니다.

하지만 (b)의 선은 실제 모집단의 분포를 70%만 제대로 분류할 수 있습니다.

(c)의 경우도 마찬가지로 일부 데이터만 샘플링해와서 라벨링하고 분류 모델을 만들게 되는데, 이 경우엔 샘플을 마구잡이로 고르는 것이 아니고 일정 기준에 따라 샘플링하게 됩니다.

이 때의 기준은 '어떤 집단에 속하는지 애매하고 헷갈리는 데이터'인데, (c) 그림상에서 볼 때 집단간 경계에 있는 모호한 샘플 위주로 추출된 것을 볼 수 있습니다.

모델이 헷갈릴만한 데이터 위주로 추출하여 정답을 알려주고 학습한다면 더 정교한 분류 모델이 만들어 질 수 있겠지요?

(c)의 모델은 모집단의 분포를 90% 제대로 분류할 수 있도록 학습할 수 있습니다.

쿼리 전략을 어떻게 정하느냐에 따라서 선별할 데이터가 달라집니다.

- 학습된 모델의 판정 값을 기반으로 뽑는 Uncertainty Sampling
- 여러 개의 모델을 동시에 학습시키면서 많은 모델이 틀리는 데이터를 선별하는 Query by committee
- 데이터가 학습 데이터로 추가될 때, 학습된 모델이 가장 많이 변화하는 데이터를 선별하는 Expected Impact
- 데이터가 밀집된 지역의 데이터들을 선별하는 Density weighted method
- 데이터들을 최대한 고르게 뽑아서 전체 분포를 대표할 수 있도록 데이터를 선별하는 Core-set approach

이 중 대표적인 두 가지에 대해 간단히 설명드리겠습니다.

### 9.2.2.1 Uncertainty Sampling

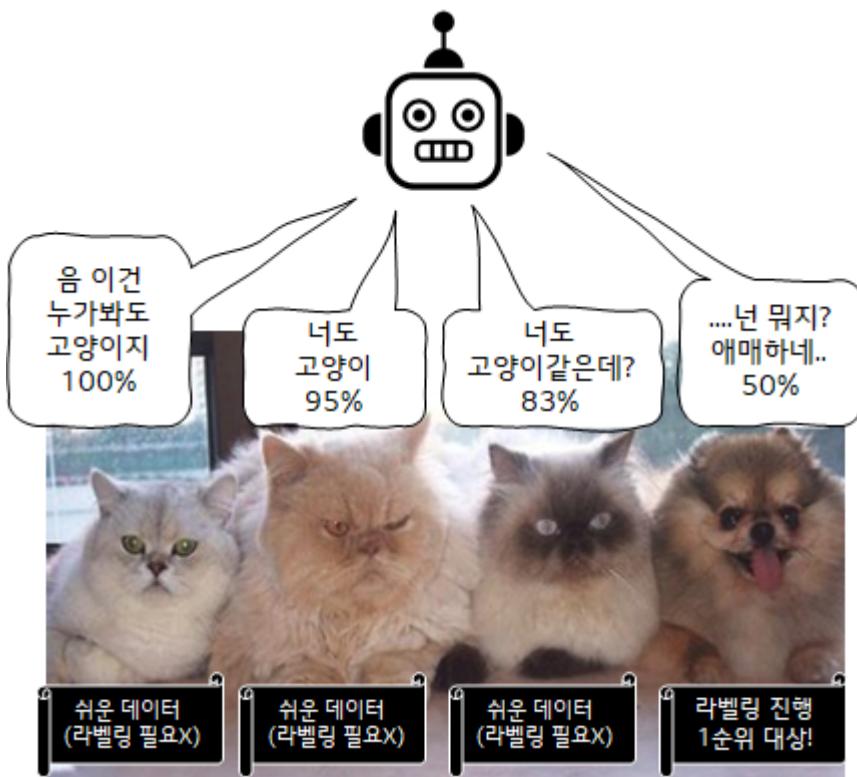
Uncertainty Sampling은 가장 단순한 쿼리 전략입니다.

AI 모델은 가장 불확실하다(least certain)고 생각하는 데이터를 추출하여 라벨링이 필요하다고 요청하게 됩니다.

예를 들어 강아지와 고양이를 분류하는 이진분류(binary classification) 태스크의 경우,

모델이 어떤 데이터에 대해 강아지일 확률과 고양이일 확률을 각각 50% 내외로 추론한다면 해당 데이터는 강아지인지 고양이인지 애매한 데이터겠죠?

이런 데이터를 라벨링하여 모델에게 알려준다면 분류 성능을 높이는 데 도움이 될 겁니다.



[그림 5. 애매한 데이터는 모델이 분류에 대한 확신이 낮을 것이다(불확실한 추론)]

#### 9.2.2.2 Query by committee

Query by committee는 여러 AI 모델간의 의견불일치를 종합 고려하는 방식입니다.

여러 모델간 추론한 결과 불일치가 많은 데이터일수록 가장 헷갈리는 데이터, 즉 라벨링을 진행할 대상이 되는 것이죠.



[그림 6. 윈:확실한 데이터, 오:애매한 데이터]

강아지과 고양이를 분류하는 모델을 여러 개 학습했다고 해봅시다.

어떤 데이터를 넣었을 때 학습한 모델간 추론 의견이 일치한다면 그 데이터는 확실히 강아지거나 고양이인 데이터일 것입니다.

그만큼 정보는 부족하겠지요. 이런 데이터는 라벨링을 진행하지 않아도 이미 여러 모델이 잘 맞춘다는 뜻이므로 넘어가도 좋습니다.

하지만 어떤 데이터를 넣었을 때 모델간 추론 결과가 제각각이라면 그 데이터는 고양이인지 강아지인지 애매한, 정보가 많은 데이터입니다.

이러한 경우 라벨링을 진행하여 모델 학습에 이용하면 분류 성능 향상에 도움이 될 것입니다.

이외에 다양한 쿼리 전략이 있습니다만, 어떤 방법이든 간에 가장 정보가 많은 데이터를 선정해서 라벨링해야 모델 학습에 도움이 될거라는 생각은 동일합니다.

예시를 통해 가장 단순한 두 방법에 대해 이해하셨다면 쿼리 전략이라는 것이 어떤것인지 이제는 아셨겠죠?

### 9.3 마무리

데이터 자체는 손쉽게 대량으로 확보할 수 있으나 모델이 학습에 사용할 수 있는 '유의미한 라벨 정보가 포함된 데이터'는 극소수입니다.

데이터는 많이 있지만, 역설적이게도 AI 모델 학습을 위해 '쓸모있는 데이터'는 많지 않습니다.

풀고자 하는 태스크를 위한 라벨 정보를 새로 만드는 것은 시간 및 비용에 의해 현실적으로 불가능한 경우가 대부분입니다.

이러한 어려움을 조금이라도 해결하기 위해 연구되어온 방법이 Active Learning입니다..

이번 주 내용을 읽어보고 느끼셨겠지만, Active Learning은 그 자체로 어떤 딥러닝 기술이라기보다는 효과적인 학습을 위한 시스템이라고 보는 것이 맞습니다.

그리고 그 시스템의 일부분에서 반드시 인간의 라벨링 작업을 필요로 하지요.

AI는 어떤 식으로든 이와 같이 조금이라도 사람의 도움(라벨링과 같은)을 필요로 합니다.

여러 사례들을 보면 Active Learning은 AI 모델 학습을 시작하는 초기 개발 단계에 매우 효과적입니다.

어차피 가르쳐줘야 할 것이라면 조금이라도 효율적으로, 더 도움이 될 수 있는 방향으로 작업을 할 수 있는 것이 좋겠죠?

그렇다 해도 Active Learning이라는 방법 하나로는 데이터에 관한 모든 문제를 해결할 수는 없습니다.

필요한 라벨 데이터의 개수가 줄어들 수는 있지만, 결국은 사람이 직접 라벨링을 수행해야 하기 때문입니다.

하지만 Active Learning은 인공지능 기술이 효율적으로 접목될 새로운 가능성을 열어줄 수는 있습니다

이번 시간은 Active Learning에 대해 설명드렸습니다.

다음 시간에는 어텐션 메커니즘(Attention mechanism)에 대해 소개하겠습니다.

감사합니다 😊

## 참고자료

- 족집게 데이터가 '전교 1등' AI 만든다!, 김도연 연구원(AI기술팀), LG CNS 블로그, <https://blog.lgcns.com/2275?category=854507>
- Introduction to Active Learning, Jennifer Prendki, KDnuggets, <https://www.kdnuggets.com/2018/10/introduction-active-learning.html>
- Active Learning tutorial, Ori Cohen, towards data science, <https://towardsdatascience.com/active-learning-tutorial-57c3398e34d>
- Active Learning Literature Survey, Burr Settles, 2010, University of Wisconsin–Madison, <http://burrsettles.com/pub/settles.activelearning.pdf>

## 10 [10편] 뭣이 중한지 알아보는 인공지능

---

안녕하세요, CTO AI빅데이터연구소입니다.

한 달에 두 번씩 **AI 테크레터**를 통해 인공지능 지식을 임직원 여러분들께 공유드리고 있습니다.

모든 CNSer가 이해하실 수 있도록 쉽게 작성하려고 하니, 상세 기술에 대한 궁금증이 생기시면 댓글이나 이메일을 통해 언제든 연락 바랍니다 😊

본 업로드는 [TECH wiki AI개시판](#)(see page 7)에서 연재됩니다.

작성 : CTO AI빅데이터연구소 AI기술팀 [김명지 팀장/총괄 CONSULTANT](#)/언어AI LAB<sup>54</sup>

---

- 긴 입력 데이터 처리하기(see page 144)
  - 어텐션 메커니즘(Attention mechanism)(see page 146)
    - 어텐션 스코어(Attention score)(see page 148)
    - 컨텍스트 벡터(Context vector)(see page 148)
  - XAI로서의 어텐션(see page 151)
    - 텍스트에서의 어텐션(see page 153)
    - 이미지에서의 어텐션(see page 153)
  - Attention 전성시대, Transformer(see page 154)
  - 마무리(see page 156)
- 

오늘은 인공신경망 모델이 특정 영역에 더 집중해서 의사결정하는 방법인 어텐션 메커니즘(Attention mechanism)에 대해 알아보겠습니다.

지난 시간까지의 내용이 궁금하신 분은 ★[AI Tech Letter](#)(see page 7)★를 확인하시기 바랍니다.

### 10.1 긴 입력 데이터 처리하기

지난시간에 배웠던 RNN에 대해 기억하시나요? RNN은 시간 순서에 따른 데이터를 처리하는 인공신경망이었습니다.([참고](#)(see page 79))

RNN의 한계는 데이터가 길어질 경우, 즉 긴 시간에 대해 누적된 데이터가 입력될 경우 먼 과거의 내용을 잘 반영하기 어렵다는 점이었는데요,

이는 사람도 마찬가지로, 사람도 최근의 내용만 기억에 남기고 먼 옛날의 사건일수록 기억이 가물가물하다는 점을 설명드렸습니다.

---

<sup>54</sup><https://wire.lgcns.com/confluence/display/~78628>

연말 평가입력을 위해 수행한 내용들을 작성할 때, 최근에 한 일에 대해서는 구체적으로 작성할 수 있지만 1,2월에 했던 내용은 어떤가요? 바로 기억이 솔솔 떠오르는 분은 잘 없으시지요?

그 때의 업무 일지나, 메일이나 플래너, 주간보고등을 다시 찾아보며 어떤 일이 있었는지 되짚어보며 평가 입력에 참고하게 되지요.

또 다른 예로, 기계번역 태스크를 생각해보겠습니다.

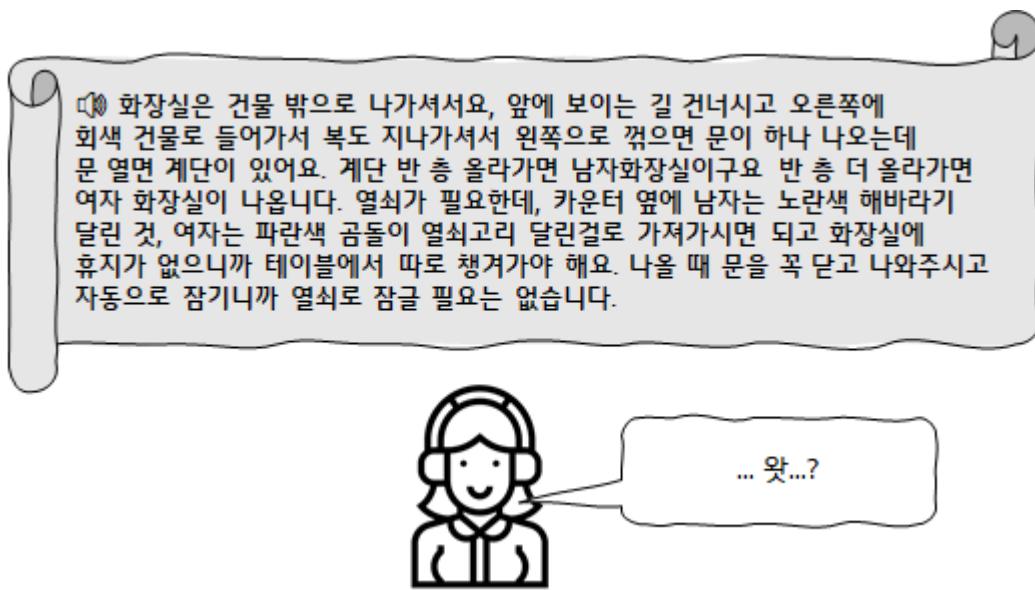
다음 문장이 음성으로 들려온다고 생각하고, 다 듣고 난 뒤 영어로 한번 번역해보시겠어요?



[그림 1. 이 정도 번역은 꺼이죠!]

참쉽죠? 여행 필수 표현으로 많이 공부했던 문장이기도 하고, 짧은 문장이기도 합니다.

그렇다면 다음을 한번 다 듣고 난 뒤 영어로 번역해야 한다고 생각해보겠습니다.



[그림 2. 저라면 이 화장실 안갑니다.]

어떤가요? 다시 들을 기회가 없다고 할 때 한번 듣는 것 만으로 전부 번역할 수 있으신가요?

내용은 어렵지 않습니다. 하지만 다량의 내용이 담긴 긴 문장(또는 문단)을 딱 한 번만 들어서는 번역을 정확하게 할 수 없을 겁니다.

우리가 긴 문장을 번역할 때엔 한 단어 또는 구절을 영어로 옮길 때마다 필요한 내용을 한국어 원문에서 되짚어가며 하겠지요.

한국어와 영어의 어순이 다르니 뒷쪽을 봤다가 다시 앞쪽을 참고하기도 하고, 여러 어절을 한 영어 단어로 번역하거나 반대로 한 어절을 여러 구로 번역하기도 할 겁니다.

문장이 문단이 되고, 문서가 될수록 더욱 더 여러 번 원문을 재참조해야 합니다.

우리의 기억력에는 한계가 있어서, 모든 인풋을 한 번 읽고 아웃풋을 한 번에 생성하는 것보다는 그때 그때 재확인하면서 필요한 부분을 보는 것이 더 도움이 됩니다.

사람이 원문(Input data)를 전체적으로 훑으며 중요한 부분을 다시 참고하듯, 인공신경망 학습 때에도 이러한 모티브를 녹여낼 수 없을까요?

## 10.2 어텐션 메커니즘(Attention mechanism)

팀 주간회의를 할 때, 내 업무 뿐 아니라 다른 팀원들의 모든 내용을 빠짐없이 다 주의깊게 들으시는 분 있으신가요? (※팀장님 제외..)

아마 대부분은 이러면 안된다는 것을 알면서도... 나와 관련된 업무 이야기를 할 때만 반짝 주의하여 듣고 나머지 시간은 노트에 낙서를 하거나 다른 생각을 하고 계시겠지요.

그러다 질문이 들어왔는데 맥락을 이해하지 못하고 어버버 거리면 팀장님의 "집중좀 합시다!!!" 한소리 하실 수도 있겠구요.



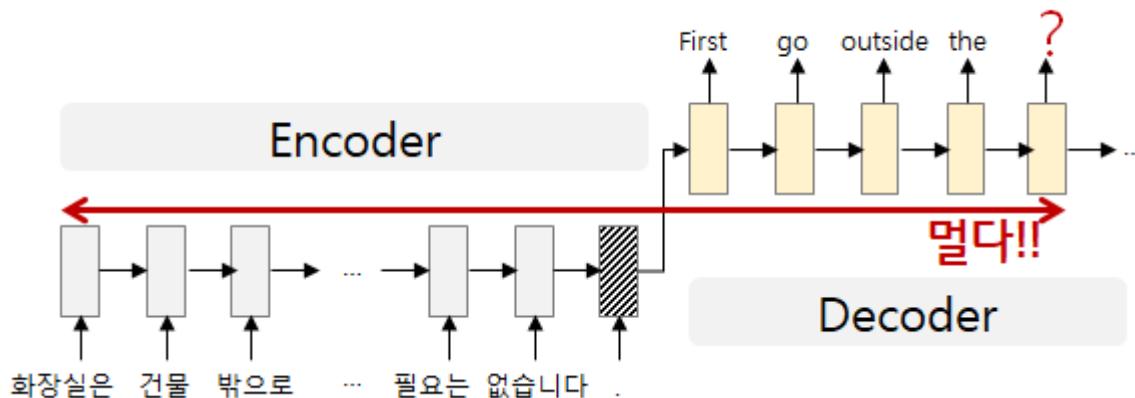
[그림 3. 다양한 과제 이야기가 오가는 팀 주간회의. 떡볶이 맛집 댓글추천받습니다.]

인공신경망 모델에서의 **집중(Attention)**이란 무엇일까요?

인공신경망이 수행하는 '집중', 어텐션 메커니즘에 대해 알아봅시다.

어텐션 메커니즘이란 인공신경망이 입력 데이터의 전체 또는 일부를 되짚어 살펴보면서 어떤 부분이 의사결정에 중요한지, 중요한 부분에 >>집중<<하는 방식입니다.

일단, 어텐션 없이 기본 RNN만으로 구성한 기계번역 모델은 아래와 같습니다.



[그림 4. 기본 RNN의 인코더/디코더를 활용한 기계번역]

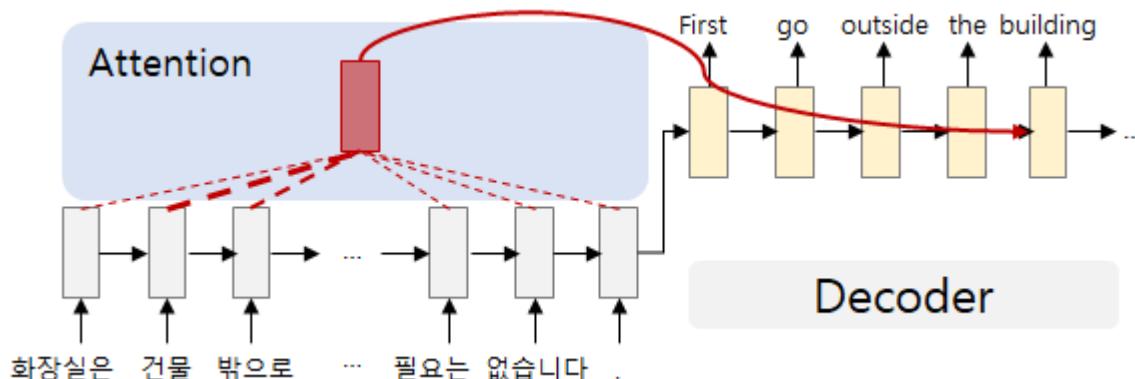
RNN은 입력 문장의 단어 하나 하나를 누적하여 압축해서 인코딩하고 있다가, 모든 문장이 다 들어오게 되면 영어로 한 단어씩 번역을 수행(디코딩)합니다.

이 때 디코더가 참고하는 문맥은 그림상의 ■에 해당하는, 입력문이 전부 압축된 하나의 벡터입니다.

이 벡터는 긴 문장을 모두 누적하고 있지만, 문장 앞부분의 내용은 너무 압축된 나머지 정보를 거의 잊어버린 것이나 마찬가지입니다.

여기서 어텐션 메커니즘을 끼얹어, 번역시에 원문을 다시 재참조하며 현재 디코딩할 단어와 연관된 중요 부분에 집중케 하면 어떨까요?

아래와 같이 구성해볼 수 있습니다.



### [그림 5. 어텐션 메커니즘을 추가로 적용한 RNN 기계번역]

"building"이란 단어를 생성할 때, 인공신경망은 전체 한국어 입력 문장을 되짚어보며 현재 단어를 디코딩하기 위해 중요한 부분이 어디일까를 생각하게 됩니다.

그 결과 수많은 입력 단어 중 "건물"에 해당하는 단어에 조금 더 주의를 기울일 필요가 있다고 판단,

해당 단어에 조금 더 집중하여 전체 입력을 다시 한번 재조정한 입력 데이터 인코딩 벡터를 만듭니다.

이렇게 하면 입력 문장이 매우 길어진다고 해도 전체 문맥을 골고루 참고할 수 있게 되므로 더 좋은 번역을 할 수 있습니다.

#### 10.2.1 어텐션 스코어(Attention score)

이 때, 중요한 단어에 집중한다는 것은 어텐션 스코어를 계산한다는 것인데요,

어텐션 스코어는 인공신경망 모델이 각 인코딩 timestep마다 계산된 특징(feature)를 가지고 자동으로 계산하는 0~1사이의 값입니다.

어떤 step은 더 집중해서 봐야하고(1에 가까운 스코어), 어떤 스텝은 지금은 중요하지 않으므로 대충대충 살피도록(0에 가까운 스코어) 하는 것이죠.

[그림 5]에서는 붉은 점선의 굵기로 어텐션 스코어를 표시해보았습니다.

각 단어에 대한 주의 집중 가중치라고 볼 수도 있겠네요.

#### 10.2.2 컨텍스트 벡터(Context vector)

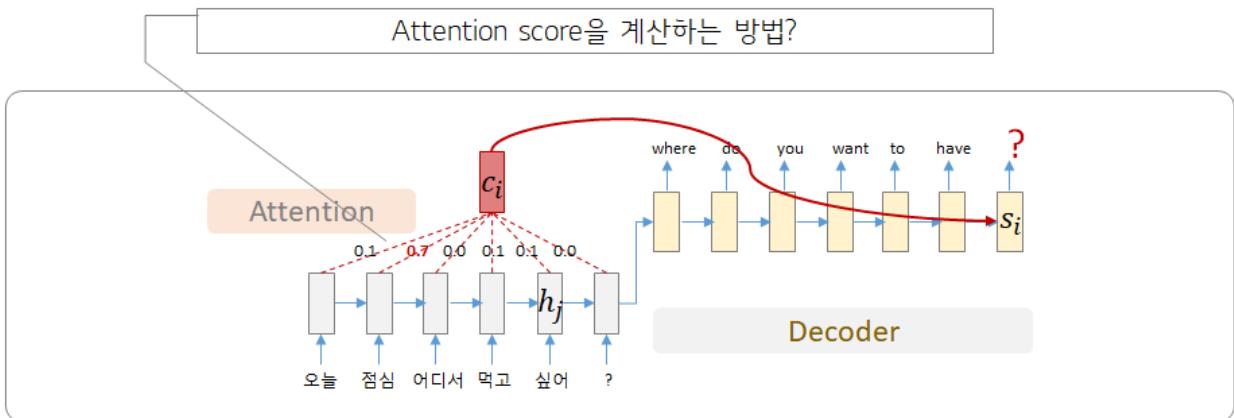
이렇게 어디를 더 살펴보고 어디는 대충 볼지에 대해 어텐션 스코어를 구하고 나면 현재 디코딩할 단어와의 관련성을 반영하여 다시 입력 문장을 인코딩하게 되는데

이는 중요도에 따라 전체 문맥의 정보를 잘 반영하고 있다고 하여 컨텍스트 벡터(Context vector)라고 부릅니다.

어텐션 스코어와 컨텍스트 벡터를 만드는 방식은 여러 가지가 있겠으나, 수학 연산 종류의 차이일 뿐이므로 깊게 다루지는 않겠습니다.

혹시 궁금하시다면 참고 자료를 봐주세요.

**참고 : RNN에서 attention score 및 context vector 계산 예 (자료:딥러닝실무과정)**



- 디코딩하는 i번째 타임스텝 직전의 hidden을  $s_{i-1}$ ,
- Attention 대상 토큰 중 j번째에 대한 hidden을  $h_j$ 라고 할 때, i번째 타임스텝의 hidden은 다음과 같이 구한다.

$$s_i = f(s_{i-1}, y_{i-1}, c_i)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}, \quad e_{ij} = a(s_{i-1}, h_j) = \begin{cases} s_{i-1}^\top h_j \\ s_{i-1}^\top W_a h_j \\ v_a^\top \tanh(W_a[s_{i-1}; h_j]) \end{cases}$$

context vector  
where  $c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$

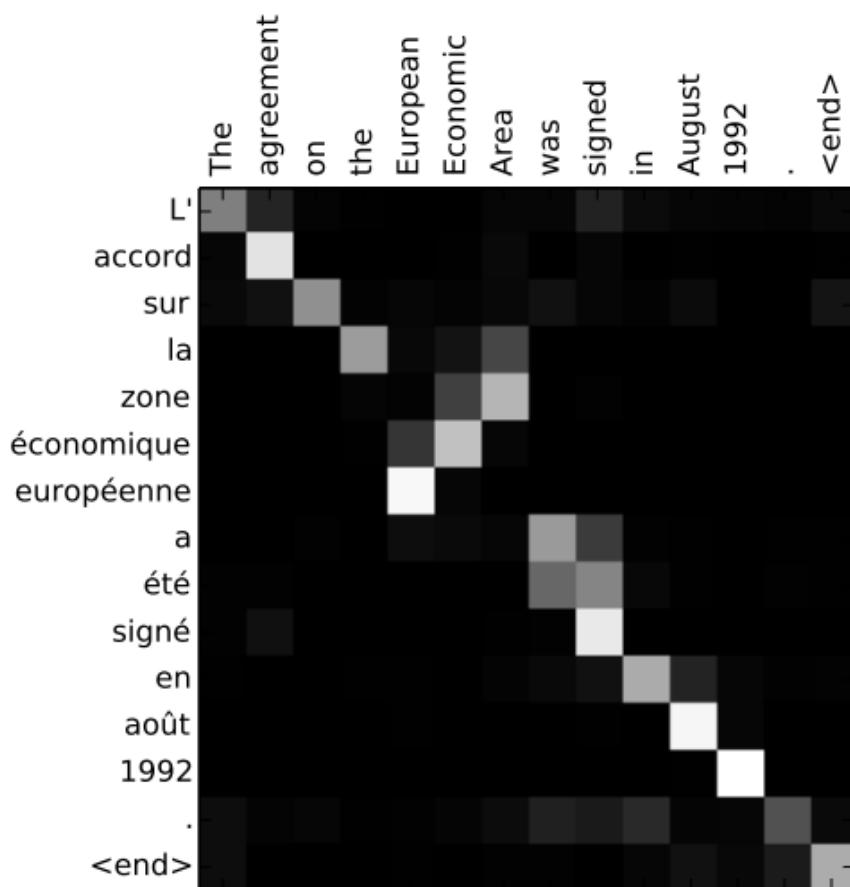
중요한 것은 스코어 계산에 필요한 수식이 아니라, 어텐션 메커니즘이 매번 디코딩마다 직전 단계의 벡터 뿐 아니라 과거의 모든 데이터의 특징(feature)들을 고려한다는 점입니다.

그리고 또 하나의 포인트는, 딥러닝 모델이 스스로 집중할 영역을 파악한다는 것이지요.

데이터를 더 잘 맞추도록 학습하는 과정에서 어떤 부분이 중요한지 사람이 알려주지 않아도 딥러닝 모델은 알아서 집중할 영역을 찾아냅니다.

딥러닝을 활용한 기계번역을 위해 어텐션 메커니즘을 처음 도입한 논문([참고<sup>55</sup>](#))에 보면 이런 자료가 있습니다.

<sup>55</sup> <https://arxiv.org/pdf/1409.0473.pdf>



[그림 6. English-French 번역에서의 attention]

위 그림은 영어 문장을 프랑스어로 번역하는 과정을 할 때, 각 단어 번역시 인공신경망이 어떤 영어 단어쪽에 집중했는지 어텐션 스코어를 시각화한 그림입니다.

하얀색으로 표시될수록 딥러닝 모델이 해당 단어를 더 주의깊게 봤다는 뜻이 되고, 이는 사람이 알려준 것이 아닌 신경망 스스로가 학습한 집중 패턴입니다.

저는 프랑스어를 잘 모르지만... 그림을 보면 영어와 프랑스어는 대강 비슷한 어순을 가지는 것으로 보이는군요!

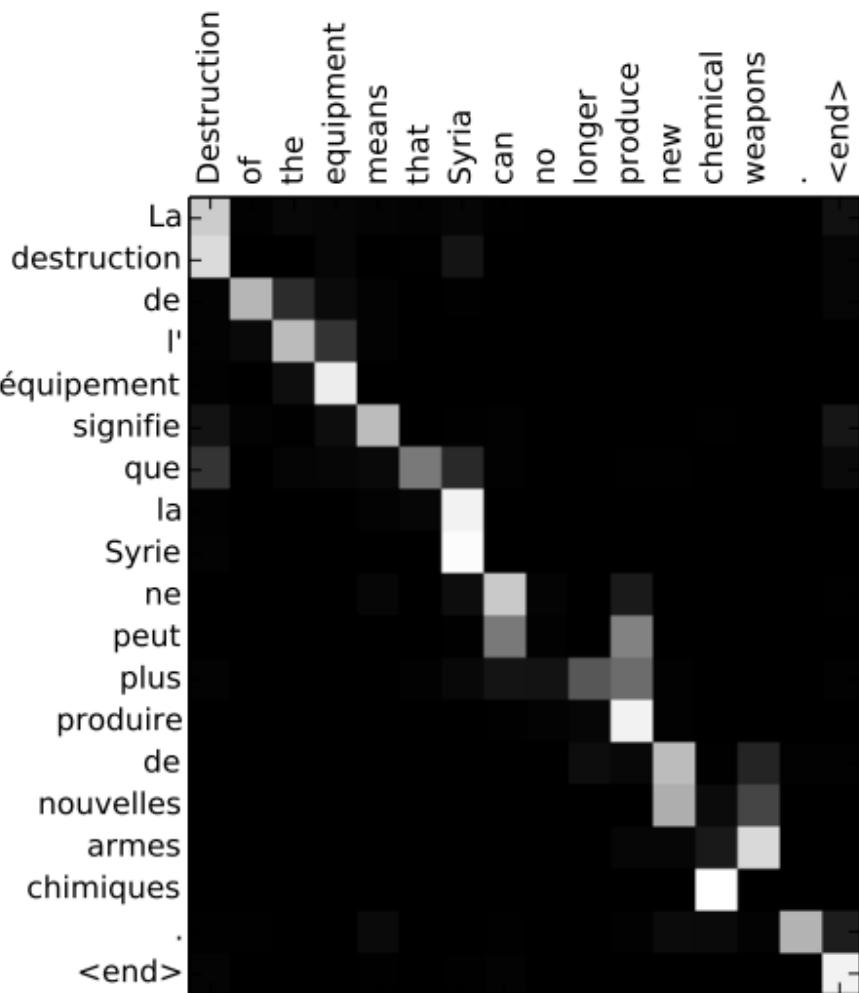
특이한점은 영어의 "European Economic Area"가 프랑스어의 "zone économique européenne"로 번역될 수 있는데, 이 부분은 영어와 프랑스어 어순이 반대입니다.

신기하게도 어텐션 스코어를 보면 딥러닝 모델도 이 부분에서는 순서를 반전시켜가며 주의를 기울이고 있네요.

완전히 동일한 어순을 가지며 단어간 1:1 매칭이 되는 언어끼리의 번역이 아니라면 그 때 그 때 유연하게 집중해야만 하는데

이 경우 어텐션 메커니즘이 그 역할을 훌륭히 해냈습니다.

하나 예를 더 살펴보겠습니다.



[그림 7. English-French 번역에서의 attention]

위 예에서 불어의 "La destruction"란 단어를 번역할 땐 영어의 "Destruction"에 집중하고, "la Syrie"를 번역할 땐 영어의 "Syria"에 집중한 걸 볼 수 있습니다.

프랑스어에서 단어 앞에 관사를 붙이는 점이 영어와 다른 특징이라고 볼 때, 해당 언어별 특징을 잘 이해하고 집중할 부분을 선택한 것을 확인할 수 있네요.

### 10.3 XAI로서의 어텐션

예제에서 알 수 있듯이 어텐션 메커니즘은 기계가 판단시 중요하게 생각하는 부분을 우리에게 알려주는 역할도 합니다.

설명가능한 인공지능(explainable AI; XAI)으로서의 기능을 수행하는데요, 이러한 영역만을 따로 연구하는 분야가 있을 정도로 인공지능의 추론 결과를 해석하는 것은 오늘날 중요한 영역입니다.

이를 해석가능한 인공지능(interpretable AI)라고도 부릅니다.

딥러닝 기반의 인공지능은 일반 머신러닝 기반이나 전통적 룰 기반의 프로그래밍에 비해 예측 정확도는 좋지만, 그 모델이 너무 복잡하고 해석하기 어렵다는 단점이 있는데요,

이러한 단점은 특히 법률, 의학, 금융 등, 민감한 내용을 다루는 도메인에서 문제가 되곤 합니다.  
인공신경망의 무수한 파라미터를 사람이 이해할 수 있는 방법으로 표현하기가 쉽지 않거든요.

예를 들어 미래에 AI 판사가 등장하여 인간 판사를 대체할 수 있을 수준이 되었다고 가정해봅시다.

AI 판사는 아주 실력이 좋은 유능하면서도 24시간 일할 수 있고, 공정하며, 뇌물수수의 유혹에도 흔들리지 않기에 인간은 전적으로 AI의 판단에 결정을 맡기게 되었습니다.

하지만 다짜고짜 AI 판사가 여러분에게 이렇게 선언했다고 해봅시다.

**AI :** "@@씨, 당신은 유죄입니다! 감옥에서 종신형을 선고합니다."

**나 :** "...? 왜... 왜죠? 저는 납득할수가 없어요! 제가 유죄라니요!"

**AI :** "이유는 설명할 수 없습니다. 하지만 제 판단은 정확합니다! 유죄! 뭔이 뭐가 필요하시다면 제 모델의 **weight**와 **bias** 파라미터들을 전부 깨보여드리겠습니다!"

**나 :** (질질끌려나가며) "이게뭐죠... 이 몇백만의 숫자들이 무슨의미가있나요...ㅠㅠ"

당연히 납득할 수 없겠죠? AI 판사가 평범한 인공신경망으로 이루어져있다면 죄명이 무엇인지, 뭘 그렇게 얼마나 잘못했길래 이런 판단을 내렸는지 설명할 수 없습니다.

반면 인간 판사는 이러한 점을 설명해줄 수 있습니다.

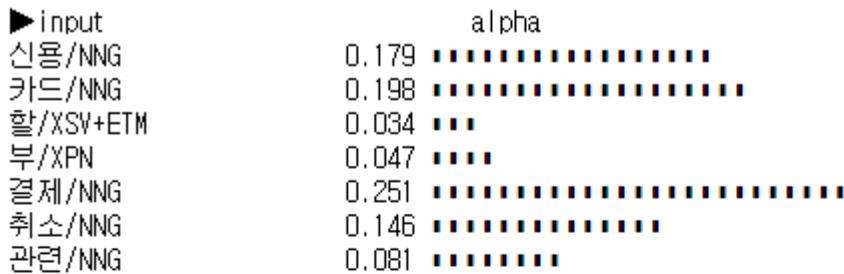
왜 이렇게 판단했는지, 죄명은 무엇이고, 어떤 점 때문에 형벌의 강도를 심하게, 또는 약하게 정했는지 등을요.

벌을 달게 받는다는게 쉬운 일은 아니지만 그래도 설명이 있다면 납득하고 받아들일 수는 있을 겁니다.

어텐션은 인공신경망의 이러한 설명 부족 문제를 일부 해소해줄 수 있습니다.

물론 우리가 알아들을 수 있는 말로 "이렇고 저렇게 때문에~ 그렇게 판단했어~"라고 알려주는 것은 아니구요, 해당 결정을 내릴 때 어떤 부분에 집중해서 판단했는지를 시각화해 보여줄 수 있습니다.

### 10.3.1 텍스트에서의 어텐션



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



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

[그림 8. 소비자 민원 주제 자동분류 예 (AI기술팀, 2017)]

위 그림은 1372 소비자민원센터의 민원 제목을 보고 딥러닝 모델로 민원 카테고리를 자동 분류하는 실험 예입니다.

민원 제목을 형태소분석한 뒤 RNN으로 주제 분류를 한 것인데, 어텐션 메커니즘을 집어넣어 모델이 어떤 키워드를 더 집중해서 보고 주제를 분류했는지 시각화해보았습니다.

첫 번째 민원에서는 '신용', '카드', '결제'와 같은 키워드에 집중해서 '금융' 관련 민원으로 분류를 했군요.

두 번째 민원에서는 '개통', '철회', '거부'에 초점을 맞추어 '정보통신서비스' 관련 민원이라는 것을 파악했네요.

만일 스마트폰 자체에 문제가 있어서 해당 키워드쪽에 더 집중을 했다면 '정보통신서비스' 보다는 '정보통신기기'쪽으로 분류했겠죠?

### 10.3.2 이미지에서의 어텐션

문장을 예로 들어 어텐션을 설명했지만, 입력 데이터의 전반적인 내용을 다시 살피며 중요한 부분에 집중한다는 사고방식은 이미지에도 적용할 수 있습니다.

아래는 Show, attend and Tell<sup>56</sup>이라는 논문에 포함된 자료입니다.



[그림 9. A woman is throwing a frisbee in a park.]

이 논문에서는 이미지 캡셔닝(Image captioning) 과제를 수행하는데, 이미지를 입력으로 주면 딥러닝 모델이 간단한 설명문을 생성하는 과제입니다.

[그림 9]의 첫 번째 사진을 입력으로 주니 인공신경망이 "A woman is throwing a frisbee in a park."이라는 문장을 생성 했네요!

뒤의 그림은 모델이 각 단어를 생성할 때 이미지의 어떤 영역을 집중해서 보았는지에 대해 어텐션 스코어를 시각화한 사진입니다.

'woman'을 생성할 때엔 사진에서 사람이 있는 부분을, 'park'를 생성할 땐 사람보다 주변의 배경에 집중하는 것을 볼 수 있습니다.

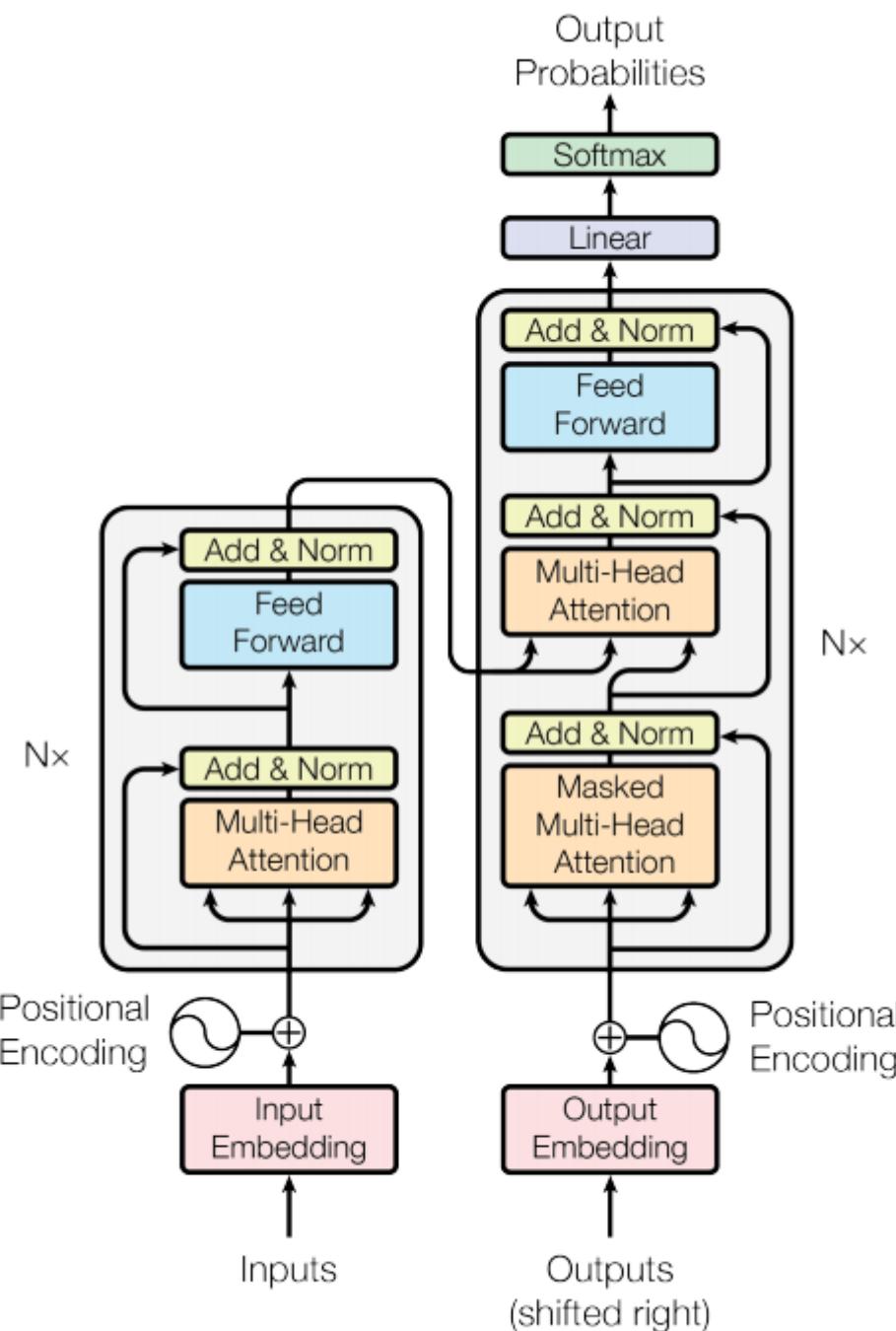
가르쳐주지 않았는데도 스스로 집중할 포인트를 찾아낸다는 게 신기하지 않나요?

## 10.4 Attention 전성시대, Transformer

이 어텐션이라는것이 어찌나 좋은지, 요즘에는 어텐션만으로 이루어진 인공신경망 구조가 새로 등장했습니다.

56 <https://arxiv.org/pdf/1502.03044.pdf>

오죽하면 그 시초가 되는 논문의 제목도 "Attention is all you need<sup>57</sup>"입니다.



**Figure 1: The Transformer - model architecture.**

[그림 10. 트랜스포머 구조]

57 <https://arxiv.org/abs/1706.03762>

트랜스포머(Transformer)라는 인공신경망은 입력 데이터끼리의 self-attention을 통해 상호 정보교환을 수행하는 것이 특징입니다.

문장 내의 단어들이 서로서로 정보를 파악하며, 나와 내 주변 단어간의 관계, 문맥을 더 잘 파악할 수 있게 되는 것이지요.

트랜스포머라는 구조는 오늘날 인공신경망 발전에(특히 자연어 이해에) 큰 획을 긋고 있습니다.

순차적 계산이 필요 없기 때문에 RNN보다 빠르면서도 맥락 파악을 잘하고, CNN처럼 일부씩만을 보는 것이 아니고 전 영역을 아우릅니다.

하지만 이해력이 좋은 대신에 모델의 크기가 엄청 커지며 고사양의 하드웨어 스펙을 요구한다는 단점이 있는데요, 이러한 한계를 보완하기 위한 다양한 경량화 방안이 연구되고 있습니다.

기회가 되면 다른 시간에 다양한 경량화 기법에 대해서도 알아보겠습니다.

## 10.5 마무리

입력 데이터의 크기가 커지면 인공지능은 정보를 효율적으로 처리하기 어렵습니다.

사람도 인공지능만큼은 아니지만, 갑자기 많은 양의 정보를 한번에 받아들이면 인식하기가 어렵죠.

하지만 사람은 데이터를 살펴보며, 어떤 부분이 중요하고 어떤 부분은 지금 당장 보지 않아도 괜찮은지 중요도를 파악하여 필요한 부분에 '집중'할 수 있습니다.

이런 모티브를 인공신경망에 녹여낸 것이 '어텐션 메커니즘'입니다.

이로써 신경망도 의사결정을 할 때 집중할 영역을 찾아 가중치를 더 반영할수 있게 되고, 또 모델의 판단을 사람이 해석하기 쉽게 만들어 줍니다.

이번 시간은 '어텐션 메커니즘(Attention mechanism)'에 대해 설명드렸습니다.

다음 시간에는 'AutoML'에 대해 소개하겠습니다.

감사합니다 😊

## 참고자료

- 자사 딥러닝 실무과정 교재보기, [교재보기] 딥러닝 실무<sup>58</sup>
- Neural Machine Translation By Jointly Learning To Align And Translate, D. Bahdanau, et al., 2015, <https://arxiv.org/abs/1409.0473>
- NLP의 궁예 등장? 관심법으로 번역을 잘해보자, J. Park, <https://jiho-ml.com/weekly-nlp-23/>
- 1372 소비자민원상담 데이터 주제분류, 2017, AI기술팀
- Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, K. Xu, et al., 2015, <https://arxiv.org/abs/1502.03044>
- Attention Is All You Need, A. Vaswani, et al., 2017, <https://arxiv.org/abs/1706.03762>

<sup>58</sup> <https://wire.lgcns.com/confluence/pages/viewpage.action?pageId=73005264>

## 11 [11편] 스스로 진화하는 인공지능, AutoML

---

안녕하세요, CTO AI빅데이터연구소입니다.

한 달에 두 번씩 **AI 테크레터**를 통해 인공지능 지식을 임직원 여러분들께 공유드리고 있습니다.

모든 CNSer가 이해하실 수 있도록 쉽게 작성하려고 하니, 상세 기술에 대한 궁금증이 생기시면 댓글이나 이메일을 통해 언제든 연락 바랍니다 😊

본 업로드는 [TECH wiki AI게시판](#)(see page 7)에서 연재됩니다.

작성 : CTO AI빅데이터연구소 AI기술팀 [김명지 팀장/총괄 CONSULTANT](#)/[언어AI LAB](#)<sup>59</sup>

---

- 사람의 손을 필요로 하는 인공지능(see page 157)
- 스스로 진화하는 인공지능, AutoML(see page 159)
  - 하이퍼파라미터 탐색 자동화(see page 160)
  - 아키텍처 탐색 자동화(see page 162)
- AutoML 특징(see page 162)
- AutoML 서비스(see page 164)
- 마무리(see page 166)

오늘은 스스로 진화하는 인공지능, AutoML(Automated Machine Learning)에 대해 알아보겠습니다.

지난 시간까지의 내용이 궁금하신 분은 [AI Tech Letter](#)(see page 7) 를 확인하시기 바랍니다.

### 11.1 사람의 손을 필요로 하는 인공지능

AI, 그 중에서도 데이터만 있다면 높은 성능을 보일 수 있는 딥러닝에 대해 어느정도 공부해 봤습니다.

이제 공부한 내용을 바탕으로 실제 AI 모델을 만들기 위해 실험을 반복할 준비가 되었는데요,

AI 사이언티스트의 길을 걷다보면 이 때부터 누구나 튜닝(Tuning)의 장벽과 마주하게 됩니다.

튜닝이라 함은 현재 실험의 결과 양상을 보고 문제점을 진단하고, AI 모델을 조금 더 나은 방향으로 만들고자 실험을 개선하는 것을 말합니다.

여기에는 아키텍처를 변경하거나, 하이퍼파라미터를 조절하거나 하는 등의 역할이 포함됩니다. ([참고:AI 테크레터 6편](#) (see page 0))

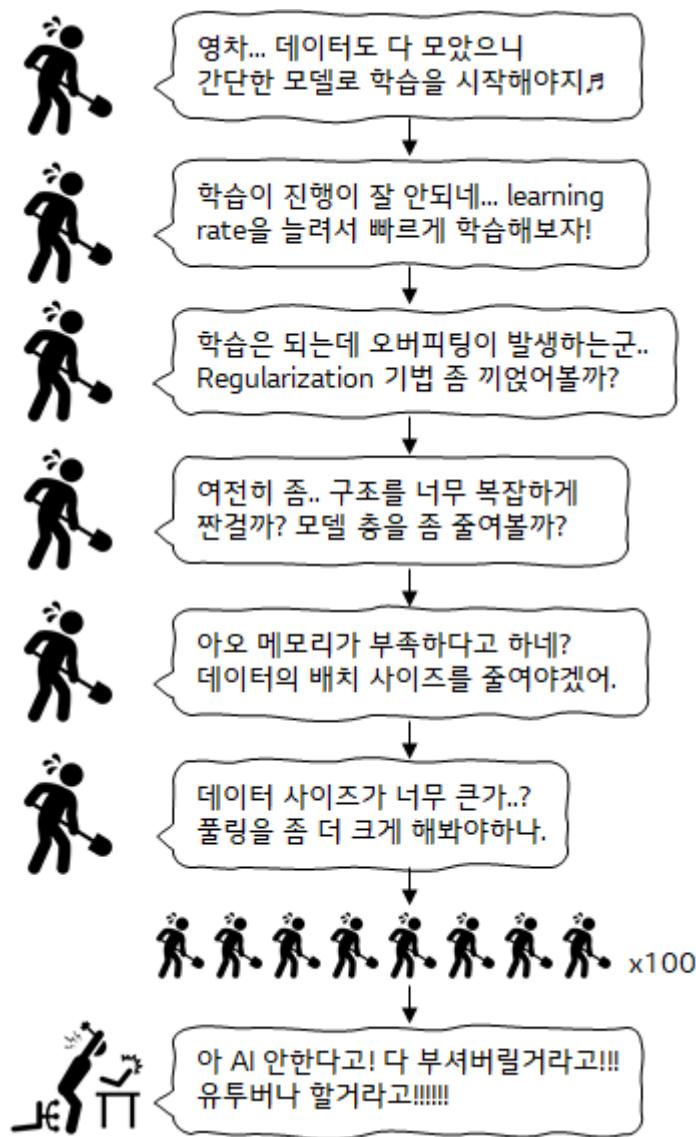
---

<sup>59</sup><https://wire.lgcns.com/confluence/display/~78628>

하지만 결과가 이럴 땐 이렇게 하는게 효과적이더라~ 하는 매뉴얼이 딱 있는 것은 아니며, 있다고 하더라도 현장의 다양한 태스크에 꼭 들어맞지 않습니다.

왜 이런 결과가 나왔는지를 해석하는 것도 사람의 뛰어난 능력 때문이 때문에 우리는 문제의 원인을 유추하고 이에 기대어 현상을 개선하기 위한 방법을 찾아야 합니다.

따라서 실험을 진행하는 사람의 탄탄한 이론 배경과 더불어 경험과 노하우까지 풍부해야만 불필요한 실험의 반복 횟수를 줄일 수 있지요.



[그림 1. AI 학습은 반복 시도와 실패의 연속. 그렇지만 반복 시도라고 해도 한계가 있다.]

우리가 튜닝해야 할 대상은 너무나 다양합니다.

딥러닝 모델로 한정짓는다 해도 FNN, CNN, RNN, Transformer... 어떤 계열의 모델 구조를 이용하는게 좋을까요?

인공신경망의 층 수는요? 얼마나 깊게 하는게 좋을까요? 한 층에 들어갈 인공 뉴런의 수(필터 사이즈, 필터 수 등등..)는요?

얼마나 성큼성큼 학습시키는 것이 좋을까요(learning rate)? 한 번에 학습할 데이터의 수는 어느 정도가 적당할까요(mini-batch size)

몇 번이나 반복해서 보여줘야 적당할까요(epoch)? 어떤 최적화 기법을 쓸 것이며(optimizer), 손실 함수는 어떤 것을 쓰는게 효과적이며(cost function), 활성화 함수는 무엇이 좋을까요?

모델 학습을 위해 사람이 결정해주어야 하는 설정이 한두가지가 아닙니다.

딥러닝 기반의 AI가 머신러닝이나 를기반의 AI보다는 수작업의 공수가 적다고 했지만, 여전히 사람의 개입이 필요한 부분이 있습니다.

조금 귀찮은데, 스스로 척 하면 척 알아서 학습할 수 있는 인공지능은 없을까요?

## 11.2 스스로 진화하는 인공지능, AutoML

다행히 우리가 수많은 시행착오를 겪으며 스트레스를 받지 않아도, 자동으로 적절한 인공지능이 학습되도록 도와주는 기법이 있습니다.

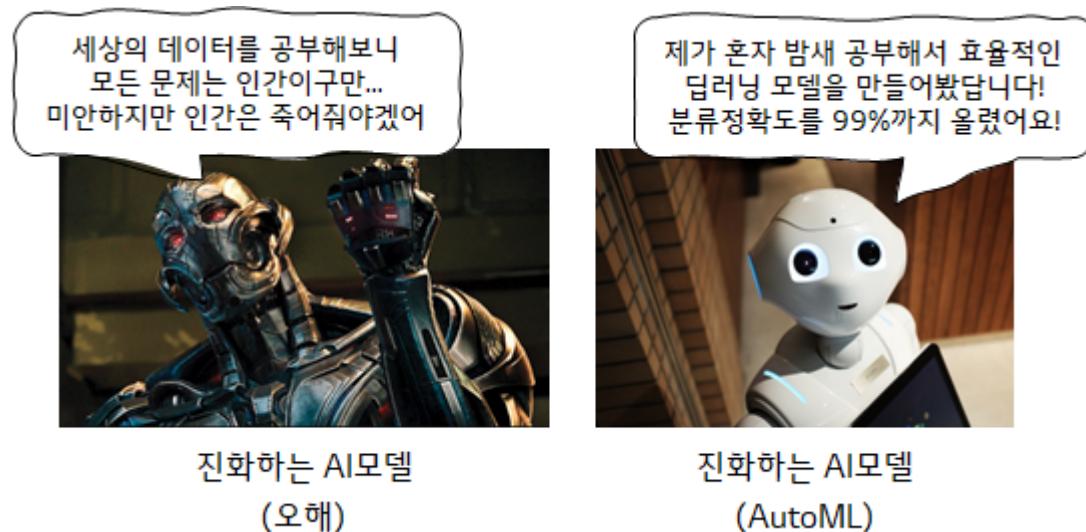
이를 **AutoML(Automated Machine Learning)**, 말 그대로 자동화된 기계학습이라고 부릅니다.

테크레터 1편에서는 인공지능이 스스로 똑똑해진다는 것은 오해일 뿐이라고 설명한 적이 있습니다. ([참고\(see page 0\)](#))

스스로 진화한다는 표현이 좀 과장되기는 했지만, 여기서 말하는 진화란 한정적인 의미입니다.

여기서는 '특정 태스크를 위한 모델 학습'에 한하여 사람이 주기적으로 실험에 개입하지 않아도 AI 스스로가 반복실험을 통해 성능을 개선하는 것을 말합니다.

가만히 놔두면 인공지능 스스로가 똑똑해져서 이것도 배우고 저것도 배우며 결국엔 사람을 지배하는 스토리의 진화가 결코 아니라는 점을 말씀드리고 싶습니다.



[그림 2. 인공지능이 스스로 똑똑해진다는 것은...]

AutoML의 역할은 크게 세 가지로 나누어 볼 수 있습니다.

첫 번째는 AI 모델을 학습하기 위해 데이터로부터 중요한 특징(feature)을 선택하고 인코딩하는 방식에 대한 Feature Engineering 자동화입니다.

두 번째는 AI 모델 학습에 필요한 사람의 설정들, 하이퍼파라미터를 자동으로 탐색해주는 것입니다.

세 번째는 AI 모델의 구조 자체를 더 효율적인 방향으로 찾아주는 아키텍처 탐색입니다.

딥러닝 모델은 비정형 데이터를 깊은 인공신경망에 태워서 자동으로 특징을 추출한다는 장점이 있기 때문에 이 종 feature engineering에 대해서는 별도로 다루지 않도록 하겠습니다.

하이퍼파라미터 탐색 자동화와 아키텍처 탐색 자동화에 대해 알아봅시다.

### 11.2.1 하이퍼파라미터 탐색 자동화

딥러닝 모델 학습에 필요한 하이퍼파라미터는 다양한 종류가 있습니다.

모델의 파라미터 업데이트를 얼마나 큰 단위로 할지를 결정하는 학습률(learning rate), 데이터를 얼마나 쪼개어 학습할지의 단위인 미니배치 사이즈(mini-batch size),

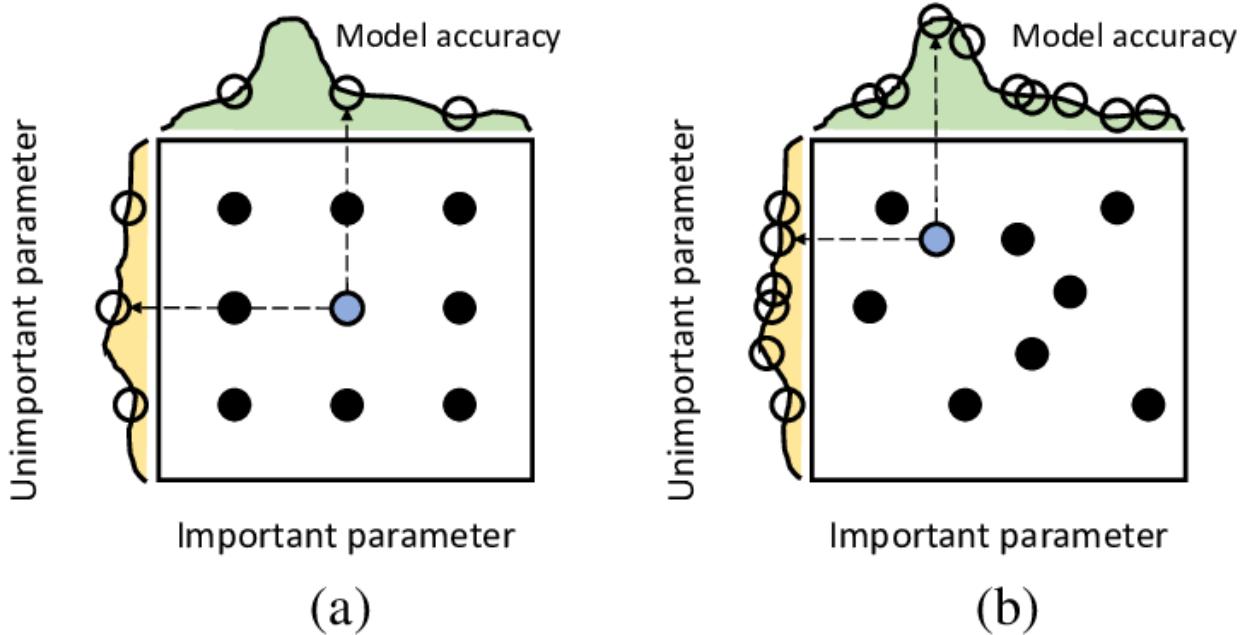
데이터를 몇 번 반복 학습할지에 대한 단위 예폭(epoch), 이외에도 모멘텀이라든지, 컨볼루션 필터의 수, 스트라이드 등등... 사람이 설정해주지 않아도 자동으로 결정되는 값은 하나도 없습니다.

많은 경우 딥러닝 학습 프레임워크(TensorFlow, PyTorch 등)에서는 기본적으로 잘 작동하는 설정을 디폴트로 제공하고 있지요.

하지만 기본 설정으로도 학습이 잘 되지 않는다면 실험 결과를 살핀 뒤 하이퍼파라미터를 조금씩 튜닝을 해줘야 합니다.

이게 워낙에 귀찮은 작업이다보니, 기존에 이미 여러 가지 하이퍼파라미터의 조합을 찾고자 하는 시도가 있었습니다.

자주 쓰이는 것 두 가지만 들자면 **그리드 서치(Grid search)**와 **랜덤 서치(Random search)** 방식이 있습니다.



[그림 3. (a)Grid search 방식, (b)Random search 방식]

그리드 서치 방식은 최적화할 하이퍼파라미터의 값 구간을 일정 단위로 나눈 후, 각 단위 조합을 테스트하여 가장 높은 성능을 낸 하이퍼파라미터 조합을 선택하는 방식입니다.

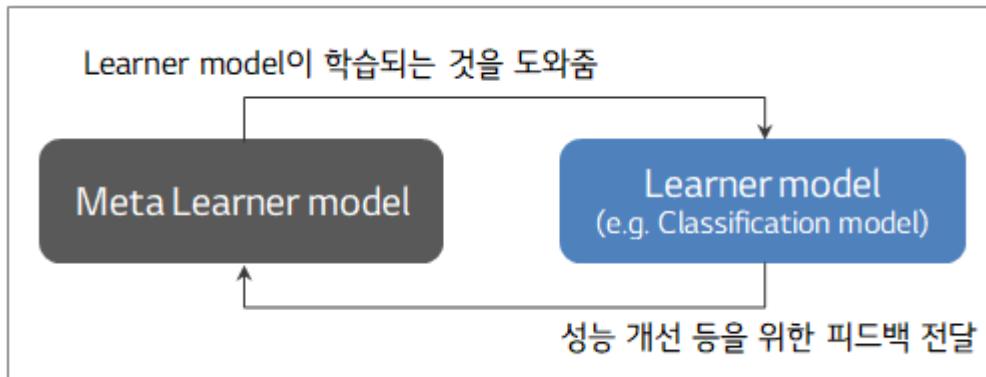
단순하지만 최적화 대상이 되는 하이퍼파라미터가 많다면 경우의 수가 기하급수적으로 많아져서 탐색에 오랜 시간이 걸립니다.

또한 불필요한 탐색에 시간을 허비하기도 하죠. 예를 들어 [그림 3]의 (a)에서 맨 왼쪽 열의 조합들은 굳이 세 번을 다 학습해볼 필요가 없지만 그리드 서치 방식을 이용할 경우 어쩔 수 없이 다 탐색을 수행해야 합니다.

반면 오른쪽의 랜덤 서치 방식은 랜덤하게 하이퍼파라미터의 조합을 테스트하는 방식인데, 그리드서치에 비해 비교적 빠르게 최적의 조합을 찾아내곤 합니다.

위의 두가지 방식은 어찌 보면 경우의 수를 찾는 단순 탐색법에 불과한데요, 단순히 for문을 이용하여 구현할 수도 있습니다.

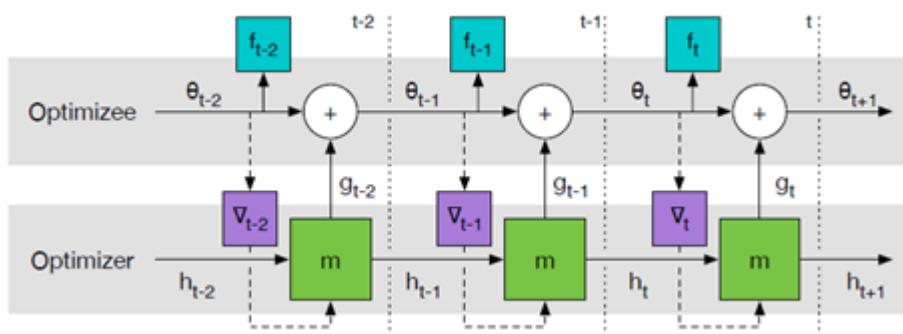
그러나 최근의 AutoML 방식에서는 하이퍼파라미터도 **모델을 통해 탐색**하곤 합니다.



[그림 4. 하이퍼파라미터를 찾아주는 Meta Learner]

주어진 태스크를 수행하는 Learner가 본래 우리가 알고 있던 AI 모델이라면, 이 AI모델이 좋은 성능을 달성하기 위한 최적의 하이퍼파라미터의 조합을 찾아주는 **Meta Learner**가 별도로 있습니다.

Meta Learner는 대부분 RNN과 강화학습을 활용하여 최적의 하이퍼파라미터를 탐색합니다.



[그림 5. Meta learner의 하이퍼파라미터 탐색]

Meta Learner의 하이퍼파라미터 조합대로 학습한 Learner의 학습 성능 결과를 Meta Learner로 다시 전달하고,

Meta Learner는 이를 또 개선하기 위한 다른 하이퍼파라미터 조합을 내며 Learner는 이 조합으로 또다시 학습하고...

이러한 과정을 반복하다 보면 최적의 조합을 찾아낸다는 이론입니다.

이 때 Meta Learner가 수행하는 학습에 대해, (Learner의)학습을 위한 학습이라는 뜻에서 '메타학습', 또는 'Learn to Learn'이라고 표현하기도 합니다.

### 11.2.2 아키텍처 탐색 자동화

하이퍼파라미터 뿐 아니라 최적의 아키텍처를 찾아주는 방법도 있습니다.

아키텍처는 모델을 이루는 구조를 말하는데, 사람이 어떤 방식으로 모델 구조를 짤지 생각하지 않아도 자동 탐색을 통해 최적 구조를 찾을 수 있습니다.

특히 딥러닝 모델의 경우에는 인공신경망을 활용하기 때문에, NAS(Neural Architecture search)라고 부릅니다.

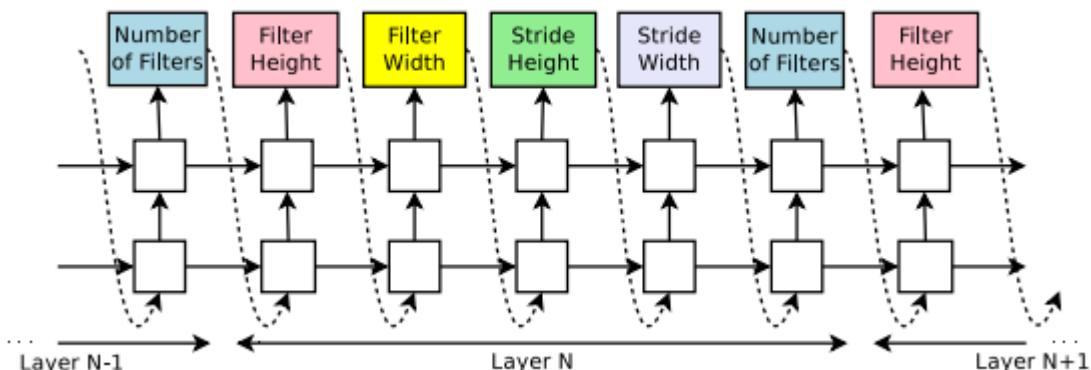
(참고: AutoML이라는 용어를 사용할 때 많은 곳에서 좁은 의미로 NAS만을 일컫기도 하지만, 넓은 의미에서 AI 자동 학습 기법이라는 측면을 더 알아주시면 좋겠네요.)

NAS의 경우도 마찬가지로 대부분 Meta Learner와 Learner로 이루어져 있어서, Learner가 본 과제를 수행하는 AI 모델이라면

Meta Learner가 어떤 구조의 신경망을 만들면 좋은지, 아키텍처 구성을 고민하게 됩니다.

Meta Learner는 역시 RNN과 강화학습을 접목한 형식으로 구성해볼 수 있습니다.

Meta Learner는 Learner의 인공신경망 아키텍처가 어떻게 구성되면 좋을지를 결정하여, Learner의 태스크 수행 결과를 보상으로 활용합니다.



[그림 6. 컨트롤러 역할을 수행하는 Meta Learner를 RNN으로 구성한 것]

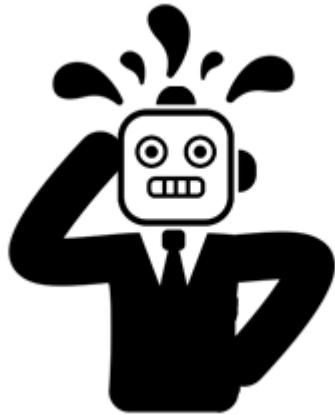
이외에도 진화 알고리즘이나 경사하강법을 기반으로 한 NAS 방식도 있습니다.

### 11.3 AutoML 특징

AutoML을 활용하면 사람이 한 땀 한 땀 구조를 고민하고, 하이퍼파라미터를 튜닝할 필요 없이 최적의 환경을 결정해줍니다.

AutoML은 일반적으로는 사람이 고민한 모델 이상의 성능을 낼 수 있다고 합니다.

기계는 사람이 생각도 못한 조합의 설정이나 구조를 시도해볼 수 있고, 기존의 설정 관습이나 제약에 얹매이지 않기 때문이죠.



[그림 7. 상상도 못한 아키텍처!]

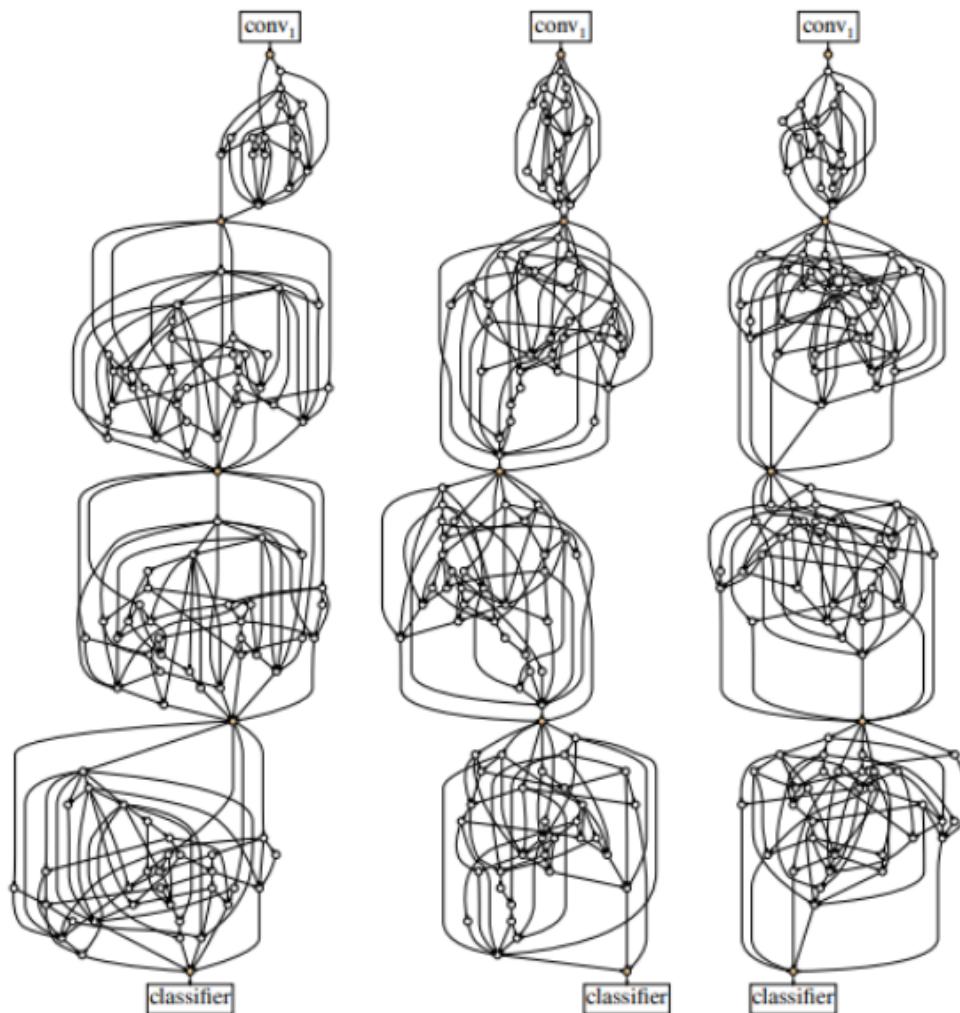
아래 그림은 작년 AI 커뮤니티를 뜨겁게 달구었던 논문의 신경망 모델인데, AutoML을 통해 자동으로 탐색된 네트워크 구조입니다. (S. Xie, et al., 2019, FAIR)

이 모델은 보기에는 볼펜똥처럼(?) 보여도 이미지 인식 과제에서 사람이 만든 신경망 모델보다 훨씬 적은 연산량으로 유사한 성능, 또는 더 좋은 성능을 보였다고 합니다.

기존의 NAS 방법들은 Meta Learner가 탐색할 아키텍처의 공간이 한정되어 있어서, 그 중 최적의 아키텍처를 찾고자 하였다면,

이 논문 저자는 이러한 제약마저 없어져야 진정한 의미의 AutoML이라고 할 수 있지 않을까? 하고 생각했다고 합니다.

이러한 모티브에서, 랜덤 그래프 생성 방법론을 기반으로 NAS를 진행하자 아래 그림처럼 기존 구성 방법의 틀을 완전히 깬 모델이 만들어졌습니다.



[그림 8. AutoML을 통해 자동으로 탐색된 인공신경망 아키텍처]

하지만 사람의 손길을 덜 필요로 하고 좋은 성능 결과를 얻는 대신, 기계가 다양한 시도를 해보도록 오랜 시간을 기다려야 합니다.

(그런데 또... 어찌 보면 사람이 반복 실험을 하며 겪는 시간보다는 적을 수도 있습니다)

또한 고사양의 하드웨어 스펙이 뒷받침 되어야만 이러한 창조적인 시도를 지원할 수 있습니다.

Learner 모델과 Meta Learner 모델이 동시다발적으로 학습을 해야 하니까요.

결론적으로 사람이 하이퍼파라미터를 찾거나 구조를 고민하기 귀찮다면 기계에게 시간과 돈을 써야 한다는 말이 되겠네요...

이런 고민을 조금이나마 덜어주기 위해 AutoML 서비스를 제공하는 업체들이 있습니다.

## 11.4 AutoML 서비스

AWS, Azure, GCP(Google Cloud Platform)와 같은 CSP 업체는 모두 일종의 AutoML 서비스를 제공하고 있습니다.

단 하이퍼파라미터 선택이나 모델 구조 선택, 이미지/텍스트 관련 태스크별 지원 기능에 차이가 있으므로 원하는 기능이 제공되는지 살펴보고 이용하시기 바랍니다.

최신 기술이니만큼 수시로 업데이트되고 있으니 당장은 제공하지 않는 기능이라 해도 향후에는 지원될 가능성이 높습니다.

CSP 서비스 중 대표적인 구글 클라우드 플랫폼은 제공하는 AutoML 기능이 가장 다양해보입니다.

시각	<b>AutoML Vision</b> 클라우드나 에지의 이미지에서 유용한 정보를 도출합니다. <a href="#">자세히 알아보기</a>	<b>AutoML Video Intelligence</b> <sup>베타</sup> 강력한 콘텐츠 탐색 기능과 매력적인 동영상 환 경을 지원합니다. <a href="#">자세히 알아보기</a>
언어	<b>AutoML Natural Language</b> 머신러닝을 통해 텍스트의 구조와 의미를 드러 냅니다. <a href="#">자세히 알아보기</a>	<b>AutoML Translation</b> 언어를 동적으로 감지하고 각 언어로 번역합니 다. <a href="#">자세히 알아보기</a>
구조화된 데이터	<b>AutoML Tables</b> <sup>베타</sup> 구조화된 데이터에서 최신 머신러닝 모델을 자 동으로 빌드하고 배포합니다. <a href="#">자세히 알아보기</a>	

[그림 9. Google AutoML 서비스]

Google AutoML은 구글 계정을 만들고 크레딧을 지불하여 이용할 수 있습니다.

제공하는 기능은 이미지에 대한 분류(classification)와 객체 탐지(detection), 동영상에 대한 분류(classification)와 객체 추적(visual tracking),

자연어에 대한 분류(classification), 객체명인식(named entity recognition), 감정분류(sentiment classification), 번역(translation),

정형 테이블 데이터에 대한 회귀(regression) 및 분류(classification)이며,

위 기능들을 엣지 레벨(엣지 기기 탑재)과 클라우드 레벨(API형식)로 제공합니다.

학습은 데이터 및 수행 과제에 따라 달라지지만, 대체로 몇 시간~1일 이내의 학습 시간을 보입니다.

비용도 상당한데요, 모든 계정에 기본으로 지급되는 무료 크레딧(100\$상당)을 활용하여 1~2회 정도의 학습을 수행할 수 있으며

비용이 걱정되는 경우 예산 커트라인을 지정하여 해당하는 만큼만 학습할 수도 있습니다.

비싸다고 생각할 수도 있지만, 사람이 직접 튜닝한다고 했을 때 코드 구현 및 시행착오에 걸리는 시간과 GPU 서버 사용료를 생각하면 더 저렴할지도 모르겠습니다.

Google AutoML로 학습된 모델은 어떤 하이퍼파라미터를 활용하며, 어떤 네트워크 아키텍처를 가지고 있는지 알 수 없습니다.

단지 우리는 비용을 지불하고, 해당 데이터에 대해 최적으로 맞춰진 모델을 이용할 수 있을 뿐입니다.

표준 데이터로 실험했을 때 대체로 사람이 학습시킨 모델과 유사하거나 더 높은 성능을 내곤 합니다.

딥러닝에 대해 잘 모르지만, 비용과 데이터가 있다면 활용해보는 것도 좋겠죠?

이외에 다양한 제공처가 있으니 나에게 맞는 서비스를 찾아보시기 바랍니다.

Tool	Platform	Input data sources		Data pre-processing	Data types detected				Feature engineering	ML Tasks	Model selection and Hyperparameter optimization		Quick start / early stop	Model evaluation / Result analysis/ Visualization				
		Spreadsheet datasets	Image, text		Numerical	Categorical	Datetime	Time-series	Other (Hierarchical types) (7)	Datetime, categorical processing	Imbalance, missing values	Feature selection, reduction	Advanced feature extraction (8*)	Unsupervised learning (10*)	Quick finding of starting model			
TransmogrifAI	Apache Spark	Y	N	Y(*)	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	N	Y	Y
H2O-AutoML	AWS, GCP, Azure	Y	N	Y	Y	Y	Y	Y	N	Y	Y	Y	N	Y	N	Y	Y	Y
Darwin (+)	GCP	Y	N	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	N	Y	Y
DataRobot (+)	AWS, GCP, Azure	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Google AutoML (+)	Google Cloud	N	Y	Y					N	Y	Y	Y	Y	Y	Y	Y	Y	Y
Auto-sklearn		Y	N	N	N	N	N	N	N	Y(2*)	Y	Y	Y	Y	N	Y	N	Y
MLjar (+)	MLJAR Cloud	Y(3*)	N	Y	Y	Y	N	N	N	Y	Y(4*)	N	N	Y(5*)	N	Y	N	Y
Auto_m1		Y	N	N	N	N	N	N	N	Y	Y	Y	Y	Y	N	Y	N	Y
TPOT		Y	N	N	N	N	N	N	N	Y	N	Y	Y	Y	N	Y	N	Y
Auto-keras		Y	Y	N	N	N	N	N	N	Y	Y	N	Y	N	N	Y	Y	N
Ludwig		Y	Y	Y(*)	Y	Y	N	Y	Y	N	Y	Y	Y	Y	N	Y	N	Y
Auto-Weka		Y	N	N	Y	Y	N	N	N	Y	Y	N	Y	N	Y	N	Y	N
Azure ML (+)	Azure	Y	Y	Y(6*)	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	Y	Y	Y
H2O-Driverless AI (+)	AWS, GCP, Azure	Y(3*)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y

[그림 10. AutoML Tool. 2019년 기준 ([자료<sup>60</sup>](#))]

## 11.5 마무리

딥러닝은 자동화의 패러다임을 바꾸었습니다.

기존의 자동화 프로그래밍이 인간의 지식을 프로그래밍 언어로 명문화하여 기계에게 주입하는 방식이었다,

딥러닝은 인간의 사고 방식 자체, 즉 인공적인 신경망을 기계에 구현한 뒤 수많은 데이터를 보여주며 말로는 표현할 수 없는 지식을 학습시켰지요.

AutoML은 여기에 다시 한 번 자동화를 추가하고 있습니다.

이제는 사고 방식이 탄생될 수 있는 환경만 조성해주면 기계가 알아서 나머지를 수행하게 됩니다.

사람은 태스크 수행을 위한 최적의 뇌 구조가 만들어질 수 있도록 지원하고 결과를 지켜보는 역할을 합니다.

60 <https://arxiv.org/pdf/1908.05557.pdf>

이번 시간은 'AutoML'에 대해 설명드렸습니다.

다음 시간에는 2020년 마지막 AI 테크레터 주제로 찾아뵙겠습니다.

감사합니다 😊

## 참고자료

- A Kernel Design Approach to Improve Kernel Subspace Identification, K. Pilario, et al., 2020, [https://www.researchgate.net/publication/341691661\\_A\\_Kernel\\_Design\\_Approach\\_to\\_Improve\\_Kernel\\_Subspace\\_Identification](https://www.researchgate.net/publication/341691661_A_Kernel_Design_Approach_to_Improve_Kernel_Subspace_Identification)
- 최신 딥러닝 기술 동향, AI빅데이터연구소 이주열 위원
- Neural Architecture Search with Reinforcement Learning, Zoph & Le, 2016, <https://arxiv.org/pdf/1611.01578.pdf>
- Learning Transferable Architectures for Scalable Image Recognition, B. Zoph, et al., 2017, <https://arxiv.org/pdf/1707.07012.pdf>
- 자동 기계학습(AutoML) 기술 동향, ETRI, 2019, [https://ettrends.etri.re.kr/ettrends/178/0905178004/34-4\\_32-42.pdf](https://ettrends.etri.re.kr/ettrends/178/0905178004/34-4_32-42.pdf)
- Exploring Randomly Wired Neural Networks for Image Recognition(FAIR), S. Xie, et al., 2019, <https://arxiv.org/abs/1904.01569>
- 박경찬 - Exploring Randomly Wired Neural Network For Image Recognition, [https://www.youtube.com/watch?v=kKaUNLrkjJM&ab\\_channel=KoreaUnivDSBA](https://www.youtube.com/watch?v=kKaUNLrkjJM&ab_channel=KoreaUnivDSBA)
- “데이터 과학자 없는 머신러닝” AutoML의 이해, itworld, <https://www.itworld.co.kr/news/129362>
- Google Cloud Platform AutoML, <https://cloud.google.com/automl>
- Towards Automated Machine Learning: Evaluation and Comparison of AutoML Approaches and Tools, A. Truong, et al., 2019, <https://arxiv.org/abs/1908.05557>

## 12 [12편] 설명 가능한 인공지능, XAI

---

안녕하세요, CTO AI빅데이터연구소입니다.

한 달에 두 번씩 **AI 테크레터**를 통해 인공지능 지식을 임직원 여러분들께 공유드리고 있습니다.

모든 CNSer가 이해하실 수 있도록 쉽게 작성하려고 하니, 상세 기술에 대한 궁금증이 생기시면 댓글이나 이메일을 통해 언제든 연락 바랍니다 😊

본 업로드는 [TECH wiki AI게시판](#)(see page 7)에서 연재됩니다.

작성 : CTO AI빅데이터연구소 AI기술팀 [김명지 팀장/총괄 CONSULTANT](#)/[언어AI LAB](#)<sup>61</sup>

---

- 종종 이해할 수 없는 결정을 내리는 AI(see page 168)
  - 설명 가능한 인공지능, XAI(eXplainable Artificial Intelligence)(see page 171)
    - XAI의 필요성(see page 171)
    - XAI를 위한 접근법(see page 172)
      - 어텐션 메커니즘(Attention Mechanism)을 활용한 XAI(see page 173)
      - 설명하는 법 학습하기(Learn to explain)(see page 174)
    - 마무리(see page 176)
    - 2020 AI 테크레터 연재를 마무리하며...(see page 177)
- 

오늘은 설명 가능한 인공지능, Explainable AI에 대해 알아보겠습니다.

지난 시간까지의 내용이 궁금하신 분은 ★[AI Tech Letter](#)(see page 7)★를 확인하시기 바랍니다.

### 12.1 종종 이해할 수 없는 결정을 내리는 AI

속을 알 수 없는 AI가 있다??

네 여기 있습니다. 인공신경망 기반의 딥러닝 모델은 사람이 그 결과를 해석하기 어렵다는 단점이 있는데요,

딥러닝 모델은 데이터가 충분한 경우 일반적으로 룰 기반의 모델이나 머신러닝 기반의 모델보다 좋은 성능을 보일 순 있지만,

어째서 모델이 이런 결과를 도출했는지에 대해 사람의 결과 해석 여지가 매우 부족합니다.

룰 기반 모델이라면 모델이 어떤 입력에 대해 정답을 맞추거나 틀리더라도

---

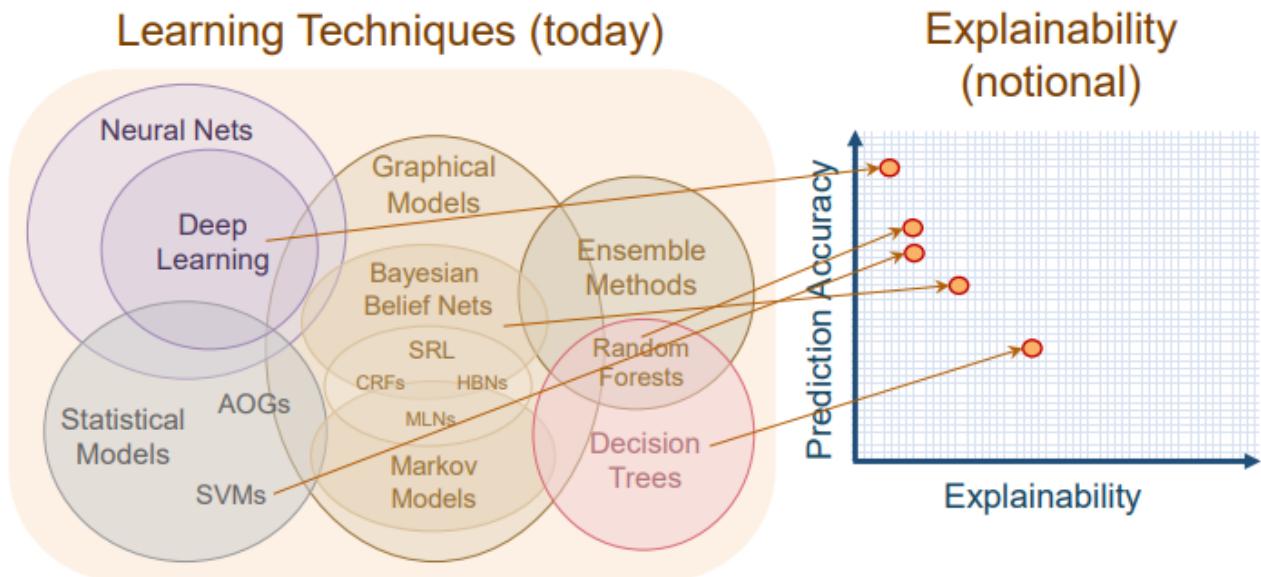
<sup>61</sup><https://wire.lgcns.com/confluence/display/~78628>

"아~ A 규칙에는 걸렸지만 B 규칙에 안맞아서 틀렸구나, 이러한 데이터도 맞출 수 있으려면 B에 예외 규칙을 좀 더 넣어야 겠어!"

같은 모델 개선방안이 나올 수 있습니다.

머신러닝 기반 모델의 경우도 마찬가지입니다.

회귀 모형이나 의사 결정 나무 같은 경우에도 어떤 입력 변수가 얼마나 영향을 미치는지, 학습된 모델을 통해 해석할 수 있는 여지가 있습니다.



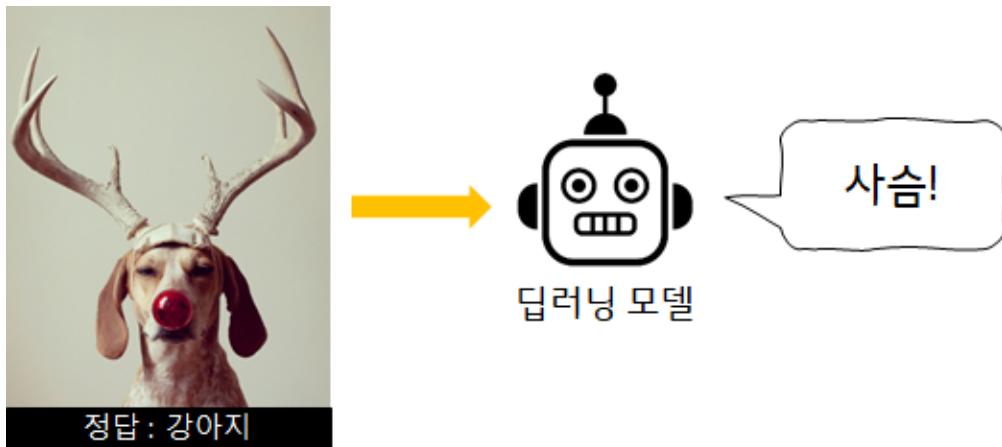
[그림 1. AI 기술별 정확도와 설명력 (자료:DARPHA)]

하지만 딥러닝 모델은요...?

규칙이나 몇 가지의 입력 변수만으로 판단하기 어려운 태스크(특히 비정형 데이터에 대한 태스크)를 수행하는 경우, 딥러닝 방식은 말로 표현하기 어려운 특징까지도 잘 포착하여 좋은 성능을 낼 수 있습니다.

수많은 파라미터로 복잡하게 얹힌 인공신경망의 연산이 우리는 뭔지 모를 어떤 판단을 잘 내리고 있는 것 같긴 한데, 간혹 "윙? 이 쉬운 걸 틀린다고?" 하는 경우가 발생합니다.

하지만 딥러닝 모델은 설명력이 부족하기 때문에 어째서 이런 결과가 나왔는지 우리는 알 수가 없습니다.



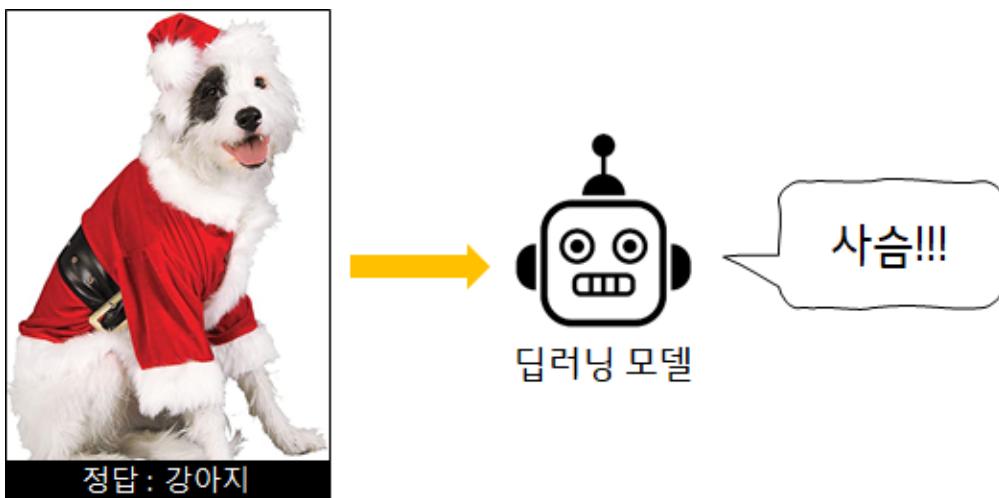
[그림 2. 딥러닝 모델의 추론 미스(1)]

그래도 위 예시같은 정도의 오답은 왜 틀렸을지 대충 짐작할 수 있습니다.

비록 모델은 사슴이라는 판단 결과 외에 아무런 추가 단서를 제공해주지 않았지만,

우리는 '그래, 뿔이 달렸으니까... 뭐 기계는 사슴이라고 생각할수도 있겠지..' 하고 정상참작 해줄 수 있습니다.

하지만...?



[그림 3. 딥러닝 모델의 추론 미스(2)]

이번엔 왜 틀린 걸까요? 어딜 봐도 사슴스러운 구석이라고는 없는데, 왜 사슴이라고 한걸까요?

차라리 산타라고 하면 모를까, 왜 산타라고는 안한거죠? 왜 강아지라고는 판단하지 않았을까요?

어떨 때 딥러닝은 잘 판단할 수 있고, 어떨 때는 실패하는 걸까요?

언제 딥러닝 모델을 신뢰할 수 있을까요?

이러한 판단 오류를 수정하기 위해서는 무엇을 더 해야 할까요?

애석하지만 위 어느 질문에도 딥러닝 모델은 대답해줄 수 없습니다.

우리는 단지 그럴싸한 가정을 찾아서 이를 해결하는 방법이라고 알려진 보완 사항을 덧붙여 모델 추가 학습을 하고, 해결이 되면 '음 그게 문제였나보네... 어쨌든 고쳐졌으니 됐는걸.' 식의 마무리를 할 수밖에요.

동물의 종류를 구별하는 정도의 과제를 수행한다면 큰 문제 없을겁니다.

하지만 이게 만일 사람의 생명을 다루는 의료 과제였다면? 법적인 문제나, 국가 보안이나, 개인/기업의 신용과 관련된 민감한 주제였다면?

사람이라면 범하지 않을 비상식적인 결과를 도출하는 AI 모델에 대해, 아무리 전반적인 성능이 뛰어나다고 해도 결과가 타당한지 신뢰하기는 어렵겠죠.

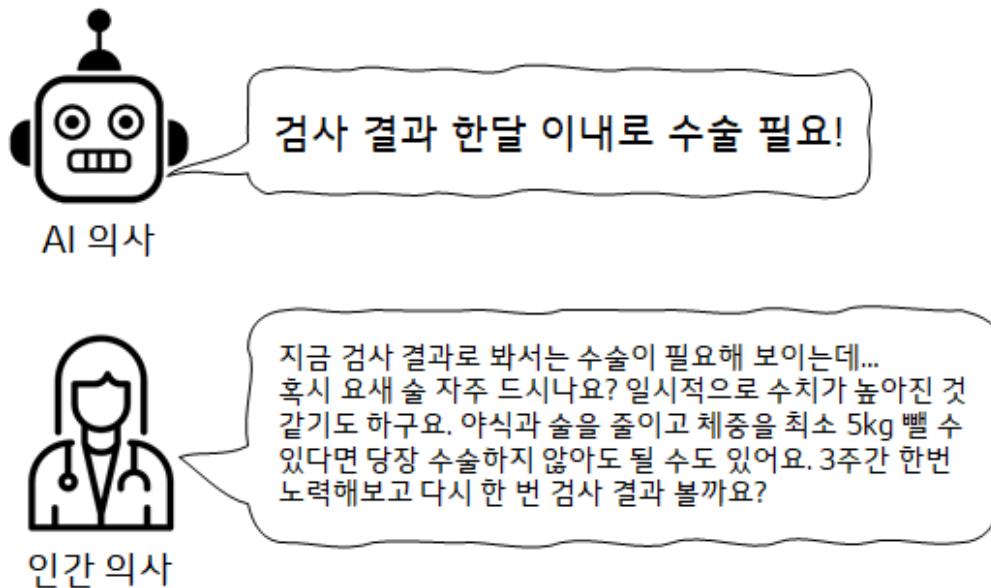
이런 인공지능의 한계를 위해, 인공지능에 설명력을 부여하는 연구 분야가 있습니다.

## 12.2 설명 가능한 인공지능, XAI(explainable Artificial Intelligence)

### 12.2.1 XAI의 필요성

AI의 판정 정확도가 사람을 능가하는 시대가 왔다고 할 때,

AI 의사와 인간 의사, 누구에게 내 운명을 맡기고 싶으신가요?



[그림 4. AI 의사와 인간 의사 (실제 사건 아님 주의)]

부연설명이 없고 단호박이지만 그래도 우수한 정확도를 보이는 AI 의사의 의견을 따르시겠습니까?

아니면 틀릴 가능성은 조금 있겠지만, 왜 이렇게 결정했는지 이유를 설명해주고 다른 결정을 내리기 위해 문제되는 부분을 지적해주는 인간 의사의 의견을 따를까요?

보통은 사람이 이해하기 어렵지만 정확도가 높은 모델 보다는 해석이 가능한 모델을 선택했을 때 더 안정감을 느낄 것이라고 생각되네요.

그렇다면 AI 의사도 "간수치가 높고 체중이 몇 KG 이상인데다 이러저러하기 때문에 수술이 필요하다고 결정했음."이라는 부연설명을 해주면 되지 않을까요?

아무리 좋은 AI 모델이라고 해도 고객사에 도입할 모델이라고 친다면,

"아 글쎄요... 이게 웬만해서는 잘 안틀리는 모델인데.. 왜 그런지는 면밀히 조사해보겠습니다(사실 조사해도 알수없음)." 보다는

"@@@하고 ###하기 때문에 \$\$\$같은 케이스에서 판단 오류가 발생했습니다. \*\*\* 처리를 이번주 중 수정하면 다시는 이런 오류가 발생하지 않을 거라 생각합니다."

라고 대처하고 싶겠죠?

인공신경망으로 구성된 딥러닝 모델은 알고리즘의 복잡성으로 인해 "블랙박스"라는 별칭으로도 불립니다.

때문에 딥러닝 모델에 설명력을 부여하는 것은 사용자가 모델의 최종 결정을 이해하고, 결과의 타당성을 확인할 수 있게 해줍니다.

즉 XAI에 대한 연구는 모델에 대한 신뢰성으로 연결될 수 있다고 말씀드릴 수 있겠네요.



[그림 5. 설명없는 딥러닝 모델...]

인공지능에게 설명력을 부여하는 방법에 대한 연구 분야를 **XAI(eXplainable AI)**, 설명 가능한 인공지능이라고 합니다.

XAI는 모델에 설명 가능한 근거와 해석력을 부여해서 투명성, 신뢰성을 확보하고자 하는 것이 목적입니다.

이렇게 되면 AI 모델의 비즈니스 활용에 있어 한계를 극복할 수 있는 가능성성이 생기겠죠?

### 12.3 XAI를 위한 접근법

XAI를 가장 본격적으로 연구하는 대표 기관에는 DARPA(Defense Advanced Research Projects Agency)가 있습니다.

DARPA는 미국 국방성의 연구 개발을 담당하는 기관으로, 2017년부터 XAI를 연구해왔습니다.

DARPA의 자료에 따르면 AI 모델을 위한 XAI로 크게 세 가지의 접근 방식이 있습니다.

첫 번째로, **기존 AI 모델에 설명할 수 있는 어떤 모듈을 덧붙이는 방식입니다.**

여기에는 다시 몇 가지 구체적인 방법이 있습니다.

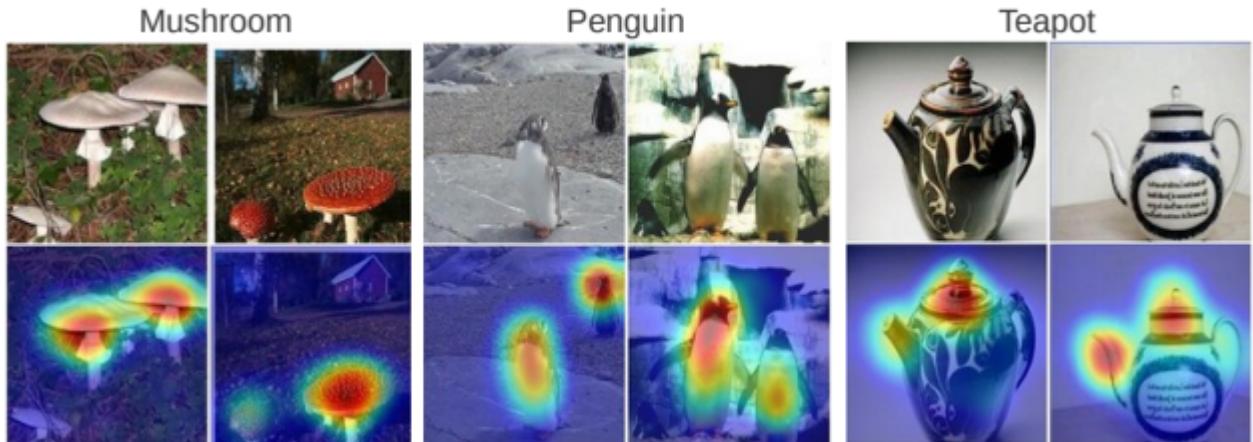
### 12.3.1 어텐션 메커니즘(Attention Mechanism)을 활용한 XAI

지난 시간의 [테크레터\(참고\)](#)(see page 144)에서 소개된 어텐션 메커니즘은 XAI로서의 기능을 수행합니다.

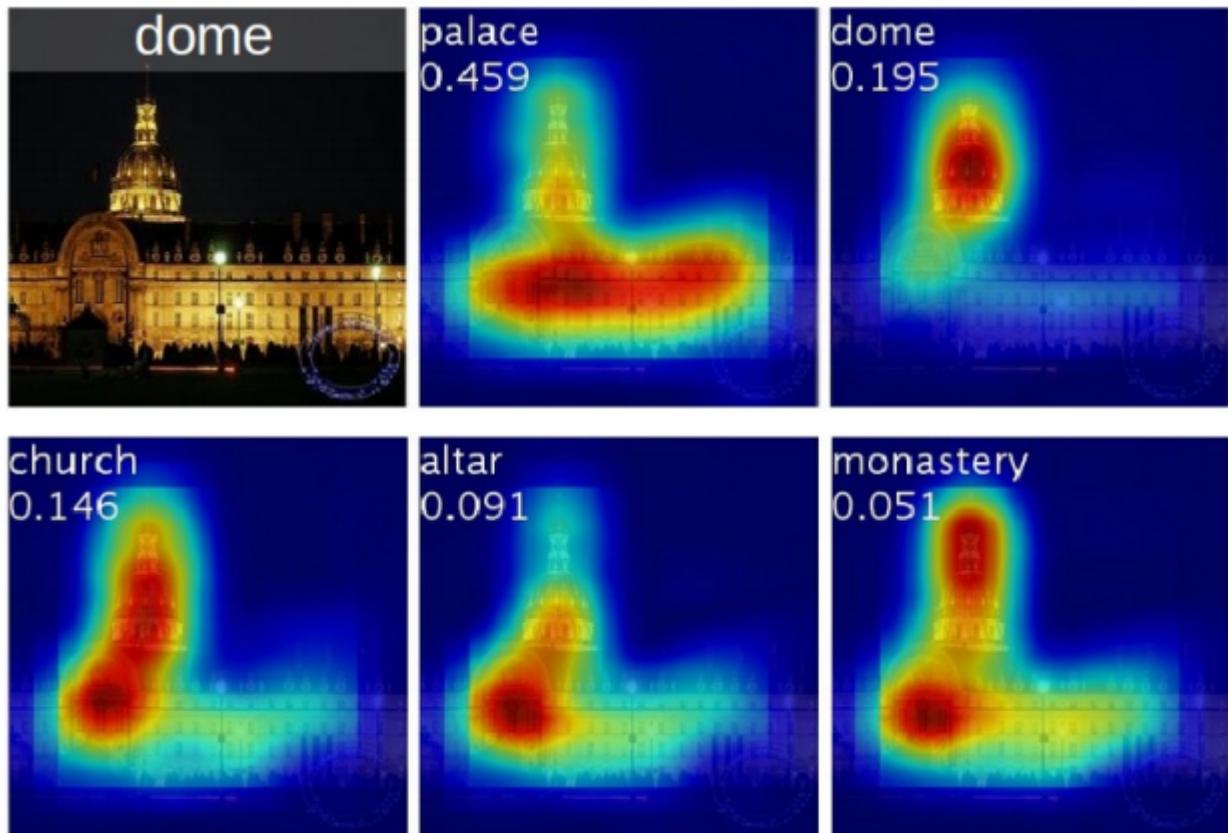
어텐션 메커니즘은 딥러닝 모델이 어떤 결정을 내릴 때, 입력 데이터의 어떤 부분에 집중해서 판단했는지를 시각화해 보여줄 수 있습니다.

집중된 영역이 반드시 판단을 내리게 된 '원인'이라고 볼 수만은 없지만,

'아 이런 부분이 중요하다고 생각해서 집중한 결과 이렇게 판단했구나~'라는 인사이트를 얻을 수 있습니다.



[그림 6. 딥러닝 모델이 버섯, 펭귄, 찻주전자로 분류한 이미지에 대해 입력 데이터의 어떤 부분을 집중했는지 시각화한 것]

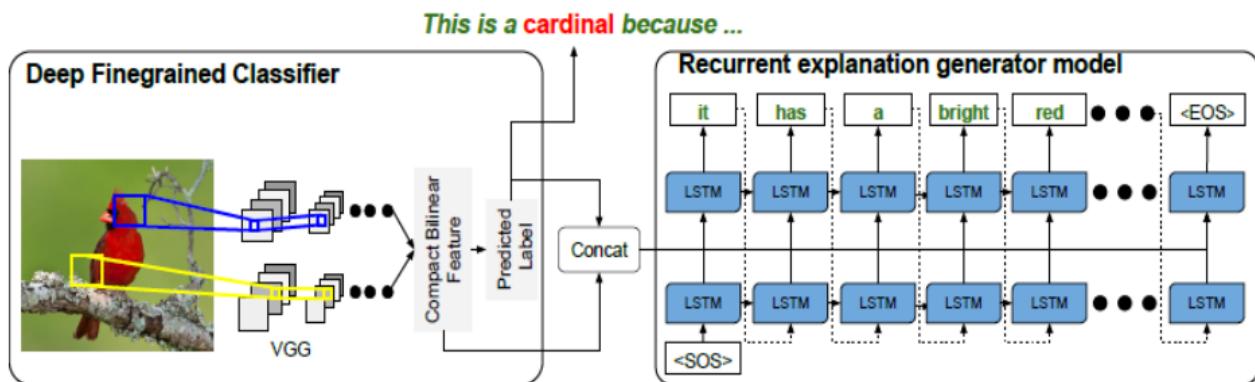


[그림 7. 첫 번째 사진에 대해, 딥러닝 모델이 분류한 상위 5개 결과에 대해 집중한 영역을 시각화한 것]

모델이 건물의 어떤 부분에 집중하는지에 따라 궁전으로도, 돔으로도, 교회로도 분류할 수 있는 점이 신기하죠?  
이러한 시각적 정보는 우리에게 AI 모델의 판단을 조금이나마 이해할 수 있게 합니다.

### 12.3.2 설명하는 법 학습하기(Learn to explain)

이 방식은 판단을 하는 딥러닝 모델에 RNN 모듈 등을 덧붙여 인간이 이해할 수 있는 방식의 설명을 생성하도록 하는 방식입니다.



[그림 8. CNN 계열의 모델이 이미지에 대해 판단하면 RNN 모듈이 설명을 생성]

이렇게 만든 모델은 어째서 이러한 판단을 결정했는지에 대해 문장을 작성할 수 있습니다.



[그림 9. 새 종류를 분류하는 모델이 판단 결과를 설명]

하지만 설명 생성에 한계가 있어 널리 활용되는 방식은 아닙니다.

이 외에도 딥러닝 모델이 해석 가능한 모듈 구성요소로 이루어진 경우, 판정 결과가 어떤 모듈 경로를 따라 연산되는지 파악하는 모듈러 네트워크(Modular Networks) 방식,

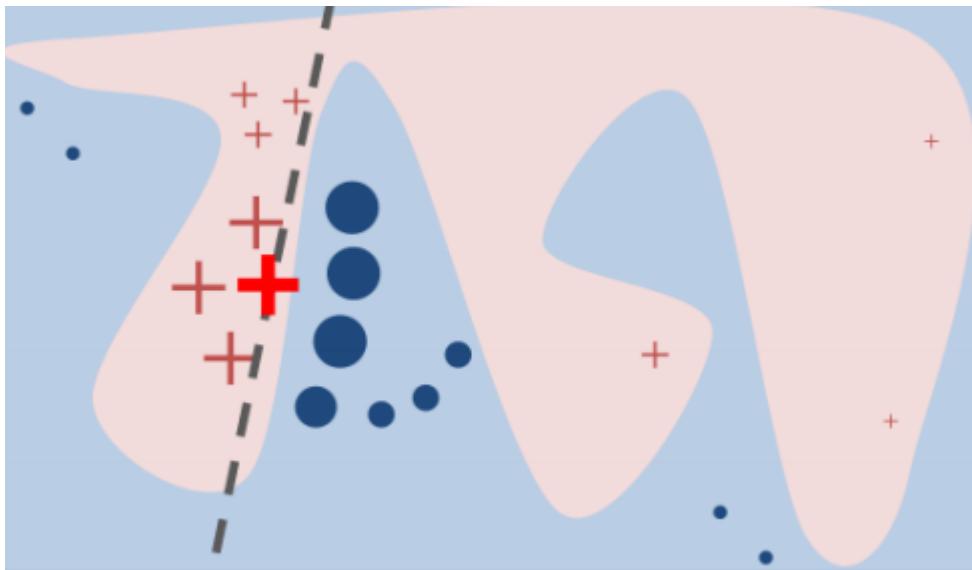
딥러닝 모델에서 “설명가능”한 특징을 학습한 노드를 찾아서 그 특징(Feature)에 ‘설명 라벨’을 붙이는 Feature Identification 방법 등이 있습니다.

두 번째로, 애초에 설명력있는 모델을 만드는 방법이 있습니다.

설명력 있는 모델의 예로는 의사결정나무나 선형회귀분석 모델 등이 있는데,

딥러닝 모델은 설명력이 부족한 인공신경망을 기반으로 하므로 이 방법을 적용할 수 없어 여기서는 따로 다루지 않겠습니다.

세 번째로는 인공신경망처럼 복잡한 블랙박스 모델의 일부분을 설명해 줄 수 있는 다른 모델을 활용하여 유추하는 방식입니다.



[그림 10. 블랙박스 모델(붉은색과 푸른색 칠해진 배경)과 설명가능한 모델(회색 점선)]

위 그림에서 연한 붉은색과 푸른색으로 칠해진 영역이 복잡한 인공신경망의 분류 결과라고 가정하겠습니다.

인공신경망은 복잡한 분류를 잘 수행했지만 설명력이 부족하기 때문에,

이 중 일부 영역의 데이터(+와 O)를 활용해 설명력이 좋은 모델을 별도로 하나 더 만들어 학습시킵니다.

그림상에서 추가로 학습한 설명력이 좋은 모델은 회색 점선에 해당합니다.

회색 점선 모델은 붉은색과 푸른색에 해당하는 전체 영역을 분류할 수는 없지만, 일부 하위 영역을 분류할 수는 있습니다.

그리고 매우 단순하기 때문에 사람이 결과를 해석하기도 쉽죠.

LIME이나 SP-LIME 등의 기술이 이에 해당합니다.

## 12.4 마무리

인간 또한 말로는 설명하기 어렵지만 '촉'이나 '감'으로 부터 결정을 먼저 내린 후 뒤이어 근거를 설명하는, 先결정 後설명을 하는 경우가 많습니다.

촉이 좋은 사람의 판단을 인정하기는 하겠지만, 그렇다고 누군가의 촉만 믿고 큰 돈을 선뜻 투자하는 등의 일은 쉽지 않을 겁니다.

구체적이고 객관적인 설명을 통해 충분히 남을 납득시킬 수 있다면 당연히 모두 설득되어 동의할 수 있겠지요.

XAI는 인공지능에게 결과만이 아닌, 과정에 대한 해석력을 부여하는 것을 중요하게 생각하는 연구 분야입니다.



[그림 11. 설명할 수 없다면 제대로 이해한 것이 아니다.]

XAI가 중요한 만큼, 여러 기술들이 연구되고는 있으나 실질적으로 좋은 성능을 유지하면서도 사람이 납득할만한 뛰어난 설명을 제공하는 방식은 아직까지는 없는 듯 합니다.

현재의 수준은 여러 가능성들 시험해보는 초기 연구 단계에 불과하다고 말씀드릴 수 있겠네요.

하지만 어느 정도 딥러닝 기반 AI 모델의 성능이 상향 평준화된 지금, 민감한 산업 도메인으로의 AI 적용 또한 활발하게 검토되고 있기 때문에

곧이어 좋은 기법들이 연구되어 높은 성능과 해석 가능성, 두 마리 토끼를 잡게 해주리라 기대합니다.

## 12.5 2020 AI 테크레터 연재를 마무리하며...

이렇게 올해 12편의 AI 테크레터가 마무리되었습니다.

어느새 우리의 삶에 부쩍 가까이 다가와 더 이상 미래의 이야기가 아닌, 당장 오늘의 큰 주제가 되는 인공지능이 단지 어려워 보인다거나 또 수학, 코딩을 잘 해야만 배울 수 있을 것 같다는 이유로 외면받는 것이 너무나 안타깝습니다. 그래서 문들이였던 제가 삽질을 하며 느리게 조금씩 터득했던 지식을, 비전공자 독자를 가정하고 조금이라도 쉽고 재밌게 전달하기 위해 노력했습니다.

혹시 부족한 저의 지식이 잘못된 내용을 전달하는 것은 아닐까, 매 회 수많은 자료를 검색하고 참고하며 작성했습니다.

딥러닝은 오픈 사이언스로, 배우고자 하는 의지만 있다면 누구나 세계 유명 대학 교수의 강의자료, 테크 자이언트 기업의 연구 논문,

그것을 구현한 코드와 데이터 등 수많은 양질의 자료에 무료로 접근할 수 있습니다.

대부분의 사람, 어쩌면 모든 사람에게 영향을 미칠 인공지능이라는 것이 어느 특정 전공자 집단이나 연구 기관의 전유물이 되지 않기를 바랍니다.

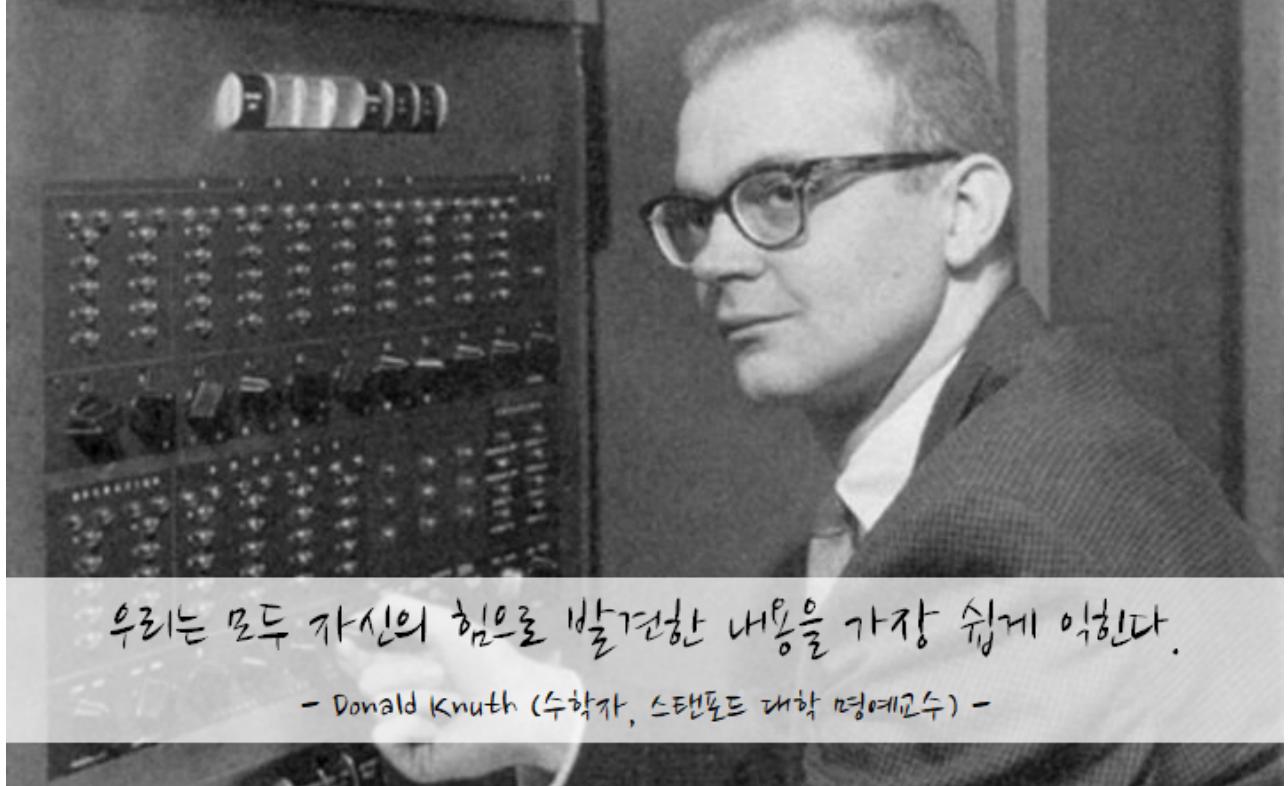
때문에 미흡한 컨텐츠임에도 불구하고 모든 임직원 여러분들께 메일을 쏴가며 읽어주시기를 호소했습니다.

부족한 부분이 있었다면 부디 너그러운 연말 직장인의 마음으로 이해해주시길 바랍니다.

AI 테크레터와 관련된 것, 또는 딥러닝과 관련된 어떠한 것이라도 질문이 생긴다면 언제든지 편하게 알려주세요.

AI 테크레터를 읽어주셔서 감사합니다.

모두 행복한 연말 보내시기 바랍니다 😊



## 참고자료

- Explainable Artificial Intelligence (XAI), David Gunning, DARPA/I2O, [https://www.cc.gatech.edu/~alanwags/DLAI2016/\(Gunning\)%20IJCAI-16%20DLAI%20WS.pdf](https://www.cc.gatech.edu/~alanwags/DLAI2016/(Gunning)%20IJCAI-16%20DLAI%20WS.pdf)
- Explainable Artificial Intelligence (XAI), Dr. Matt Turek, <https://www.darpa.mil/program/explainable-artificial-intelligence>
- 딥러닝 이후, AI 알고리즘 트렌드, ETRI Insight Report, 2018
- 설명 가능한 인공지능(eXplainable AI, XAI) 소개, 금융보안원 보안기술연구팀, 2018
- 위키피디아-방위 고등 연구 계획국, <https://ko.wikipedia.org/wiki/%EB%B0%A9%EC%9C%84%EA%B3%A0%EB%93%B1%EC%97%B0%EA%B5%AC%EA%B3%84%ED%9A%8D%EA%B5%AD>
- Class Activation Mapping and Class-specific Saliency Map, <http://cnnlocalization.csail.mit.edu/>
- Title: Top-down Visual Saliency Guided by Captions, V. Ramanishka, et al., 2017, <https://arxiv.org/abs/1612.07360>
- Learning Deep Features for Discriminative Localization, B. Zhou, et al., 2015, <https://arxiv.org/abs/1512.04150>
- AI와 최신 딥러닝 기술 동향, AI빅데이터연구소 이주열수석위원, 2019, [http://delab.cju.ac.kr/seminar/LG\\_ai.pdf](http://delab.cju.ac.kr/seminar/LG_ai.pdf)
- Generating Visual Explanations, L. Hendricks, et al., 2016, <https://arxiv.org/abs/1603.08507>
- "Why Should I Trust You?": Explaining the Predictions of Any Classifier, M. Ribeiro, et al., 2016, <https://arxiv.org/abs/1602.04938>