

AI와 최신 딥러닝 기술 동향

2019.10.29

AI빅데이터연구소
이주열 연구소장



목 차

- I. AI 정의
- II. 딥러닝 기술 Stack
- III. 딥러닝 알고리즘 동향
- IV. 딥러닝 F/W 기술 동향
- V. 딥러닝 적용 사례 및 조언



AI 정의

I. AI 정의

▪ 인공지능(Artificial Intelligence) 교과서¹⁾가 정의하는 인공지능

- (1) 인간처럼 생각하는 시스템

예: 인지적 구조와 신경망

- (2) 인간처럼 행동하는 시스템

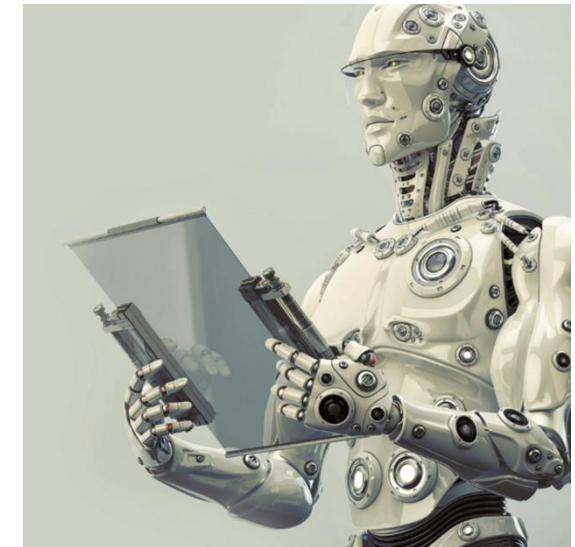
예: 언어 처리, 지식 표현, 자동 추론, 학습으로 Turing시험 통과

- (3) 합리적으로 생각하는 시스템

예: 논리풀이, 추론, 최적화

- (4) 합리적으로 행동하는 시스템

예: 지능적인 소프트웨어 에이전트, 인식, 계획, 추리, 학습, 대화, 의사결정, 행동 로봇 구현



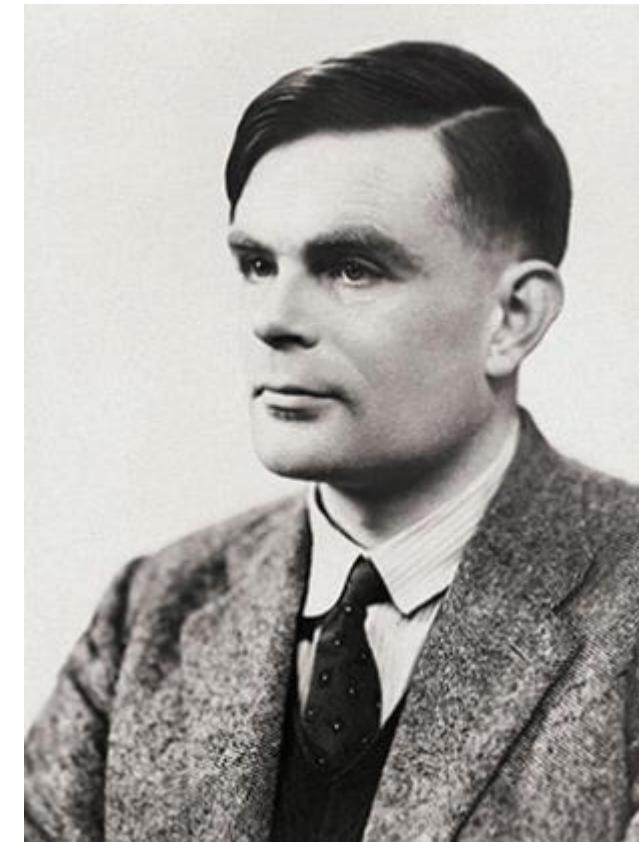
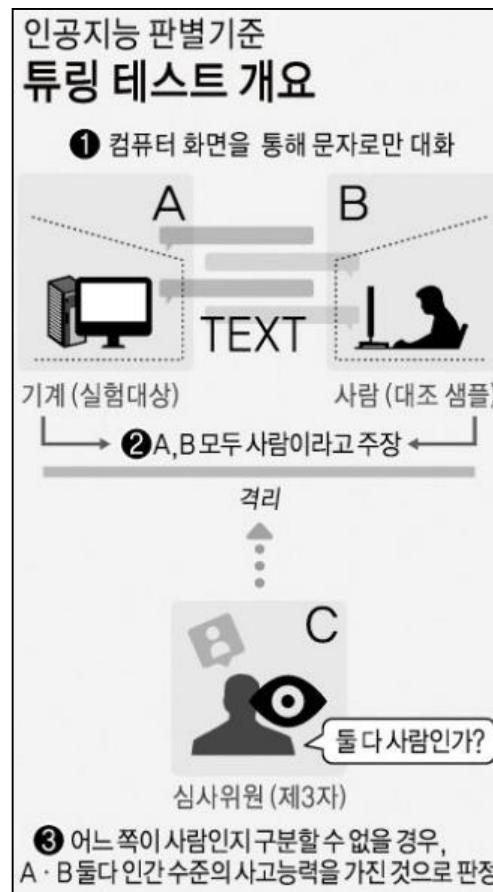
주1) Artificial Intelligence: A Modern Approach - Stuart Russell & Peter Norvig

I. AI 정의

▪ 인공지능 판단 기준

- Turing 테스트¹⁾

: 컴퓨터로부터의 반응을 인간과 구별할 수 없다면 컴퓨터는 생각(thinking)할 수 있는 것

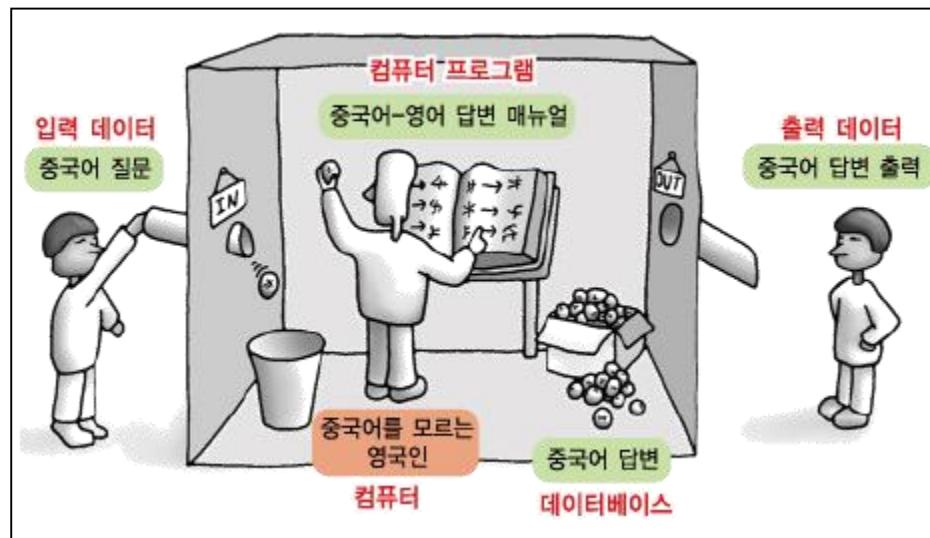


주1) Turing Test : Alan Turing이 1950년에 철학저널 Mind에 발표한 Computing Machinery and Intelligence에서 제안된 방법

I. AI 정의

▪ 인공지능 판단 기준

- 중국어 방(Chinese Room)에 대한 논증¹⁾: Turing 테스트로는 인공지능을 판단할 수 없다
 - . 방 안에 있는 존재는 실제로는 중국어를 전혀 모르며 중국어 질문을 이해하지 않고 주어진 표에 따라 대답할 뿐이라면, 중국어로 질문과 답변을 완벽히 한다고 해도 안에 있는 존재가 중국어를 진짜로 이해하는지 어떤지 판정할 수 없다는 결론



Rule이 인공지능인가?

주1) John Searle의 Turing Test로 인공지능 여부를 판정할 수 없다는 것을 논증하기 위해 고안한 사고 실험

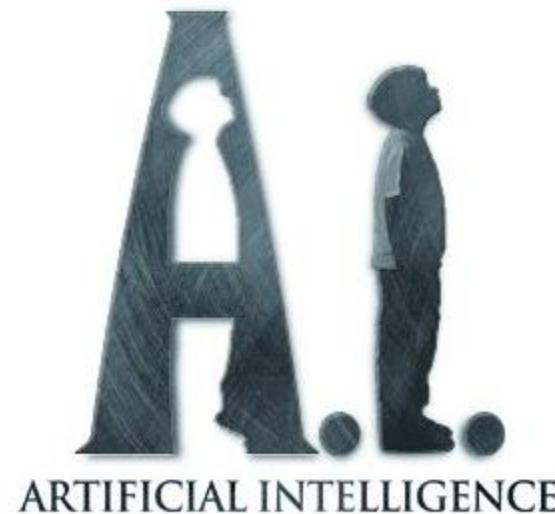
I. AI 정의

- 그래서... 인공지능의 정의는?



모든 전문가들이 동의할 만한 인공지능의 정의는 이 세상에 존재하지 않음.

다만 대략 “지능적인 행동의 자동화”를 표방함.



I. AI 정의

▪ AI 교과서¹⁾가 정의하는 인공지능

- (1) 인간처럼 생각하는 시스템

예: 인지적 구조와 신경망

**Strong AI
(General AI)**

- (2) 인간처럼 행동하는 시스템

예: 언어 처리, 지식 표현, 자동 추론, 학습으로 Turing시험 통과

- (3) 합리적으로 생각하는 시스템

Weak AI

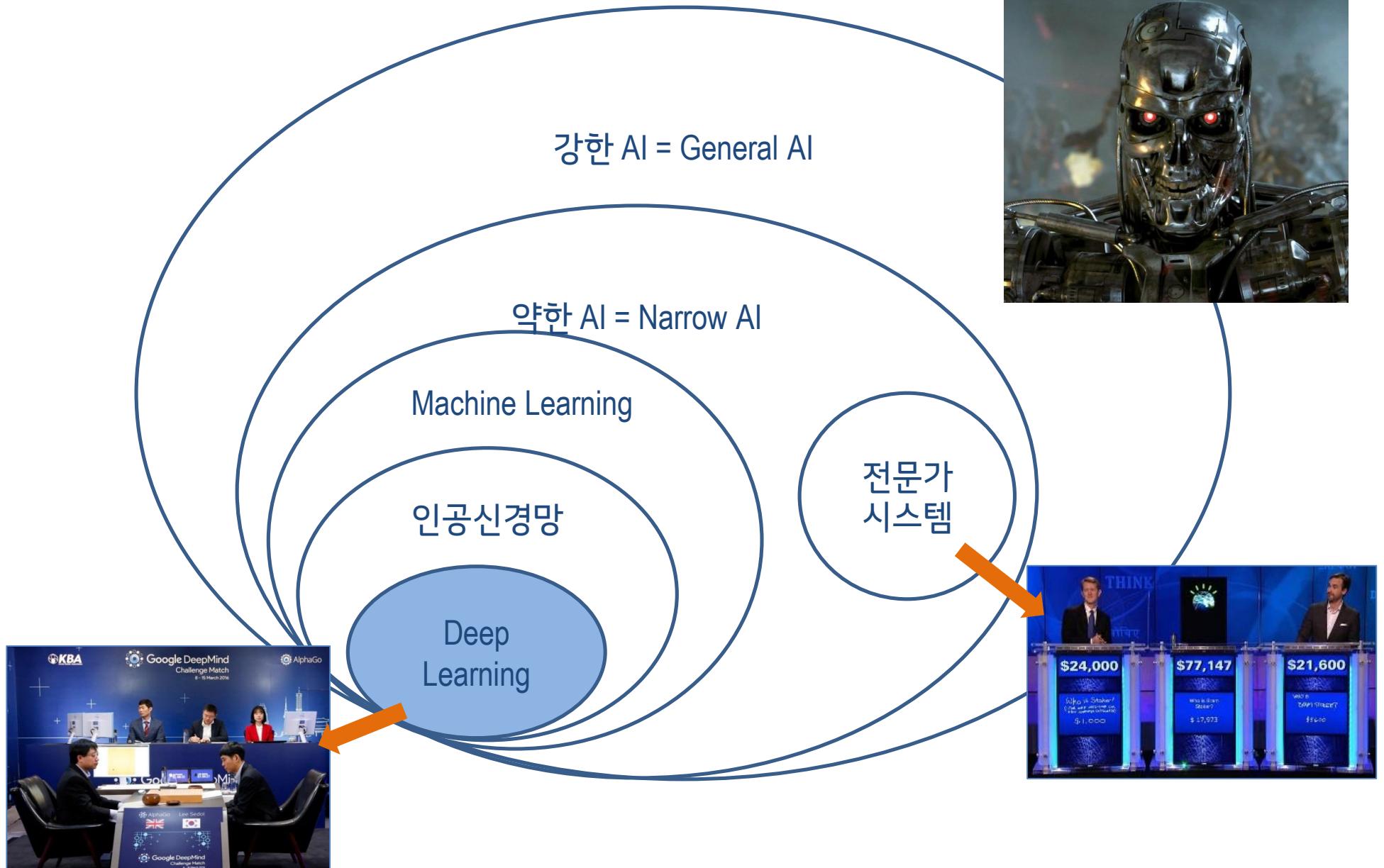
(Narrow AI)

- (4) 합리적으로 행동하는 시스템

예: 지능적인 소프트웨어 에이전트, 인식, 계획, 추리, 학습, 대화, 의사결정, 행동 로봇 구현

주1) Artificial Intelligence: A Modern Approach - Stuart Russell & Peter Norvig

I. AI 정의

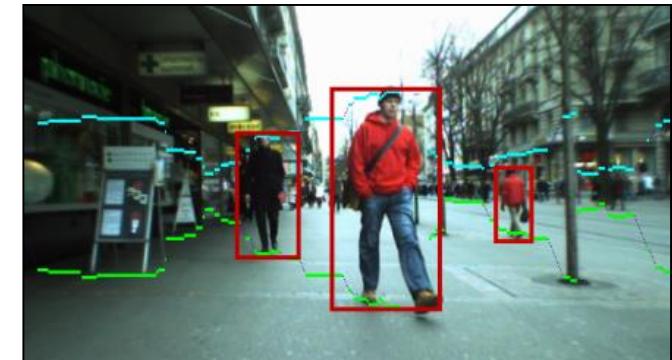
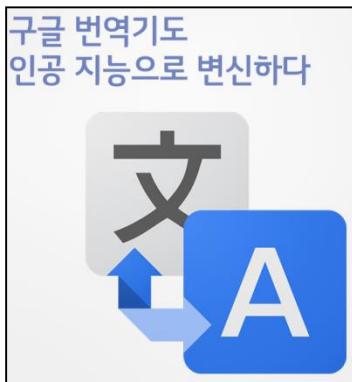


I. AI 정의

▪ AI의 현재

- Weak AI(Narrow AI): 번역, 자율주행, 영상인식 분야에서 놀라운 발전 → 현재
- Strong AI(General AI¹⁾): 인간과 동일한 수준의 지능적 행동을 구사하는 미래 인공지능 → 미래
 - . 미국 국가과학기술위원회(NSTC)는 당분간 General AI의 등장은 힘들 것으로 판단
- 그러나, Strong AI는 AI 연구의 지향점

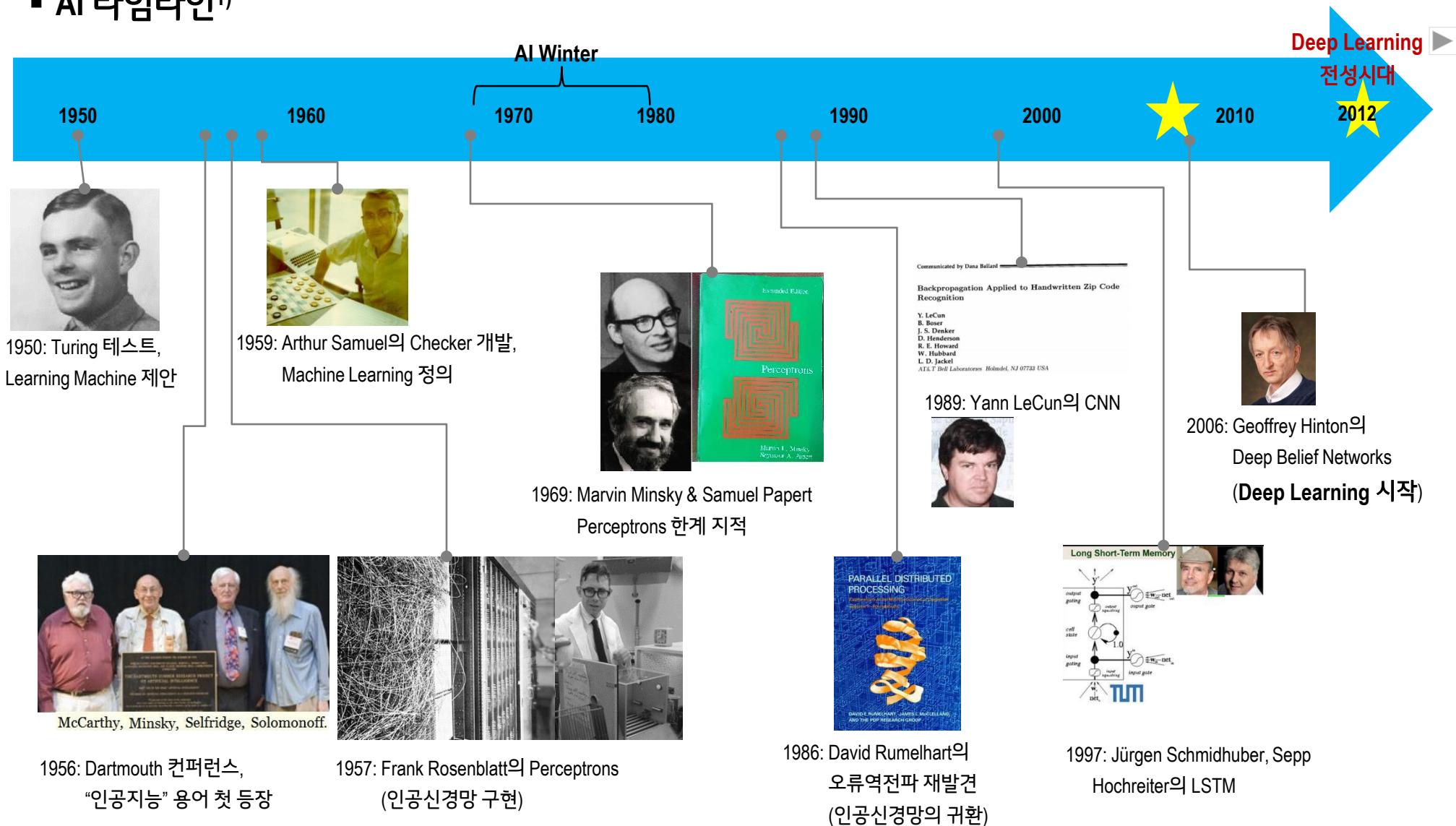
▪ 현재는 필요한 문제를 해결해 줄 수 있는 실용적인 Weak AI 개발 및 활용 중심



주1) 또는 AGI(Artificial General Intelligence)라고 부름

I. AI 정의

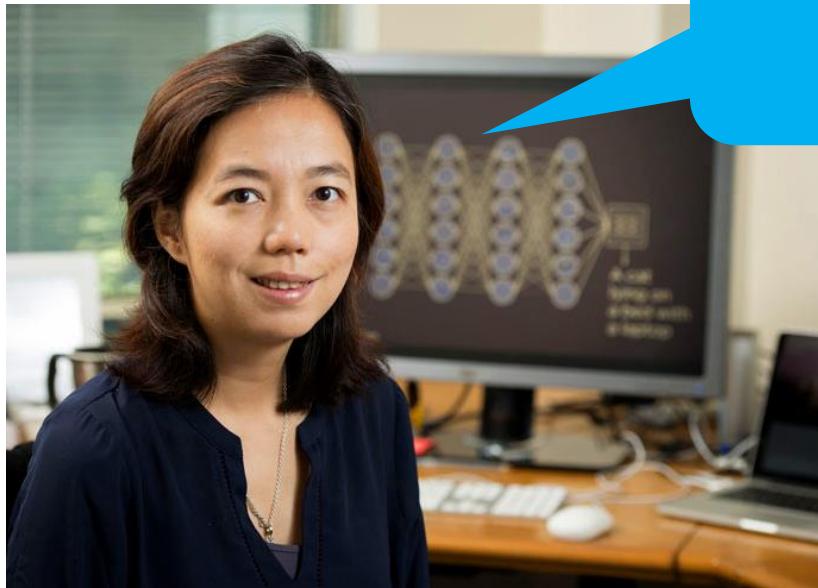
■ AI 타임라인¹⁾



주1) 위키피디아 Timeline of Artificial Intelligence 참조: https://en.wikipedia.org/wiki/Timeline_of_artificial_intelligence

I. AI 정의

▪ 그러나.... AI 현재 한계



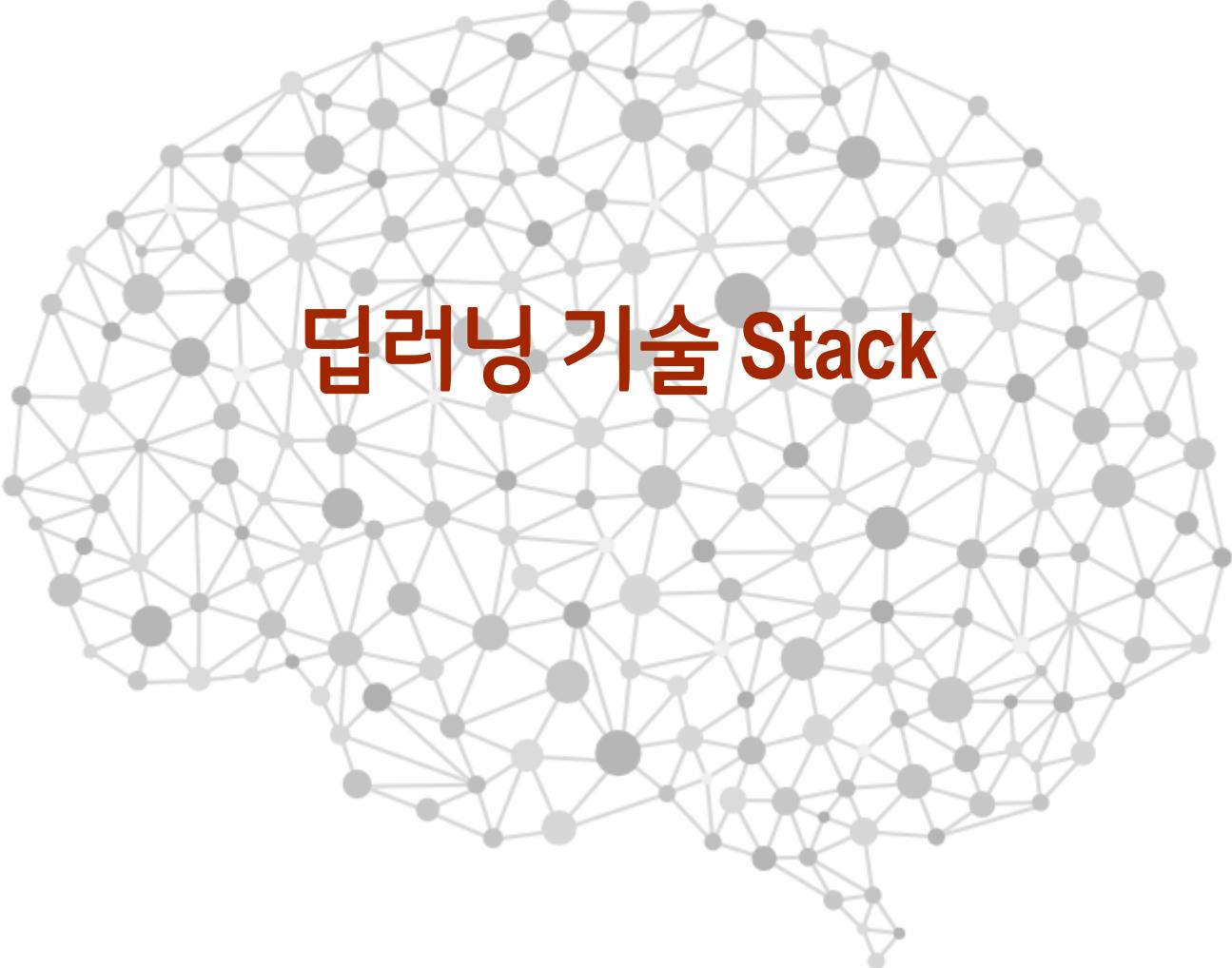
“현대 AI는 불이 난 집에서
완벽한 바둑 한 수를 두는
기계를 의미한다.”

범용성

맥락 이해

Fei-Fei Li 교수

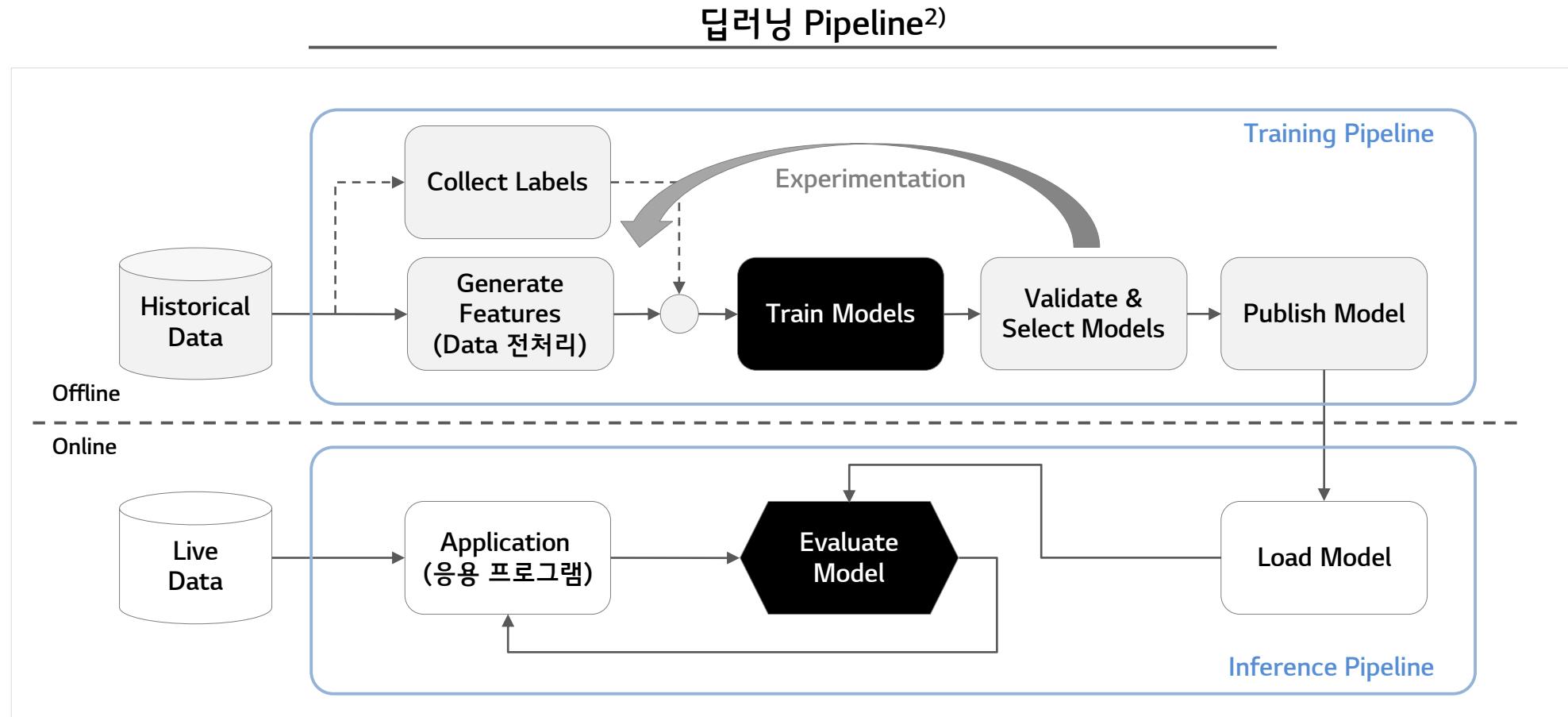
- Stanford Univ. 컴퓨터과학과 교수
- Stanford AI Lab (SAIL) 디렉터
- ImageNet 프로젝트 창시자
- Google Cloud Platform AI 수석 과학자



딥러닝 기술 Stack

I. 딥러닝 기술 Stack

▪ 딥러닝¹⁾ 모델 개발/사용을 위한 프로세스(Pipeline)



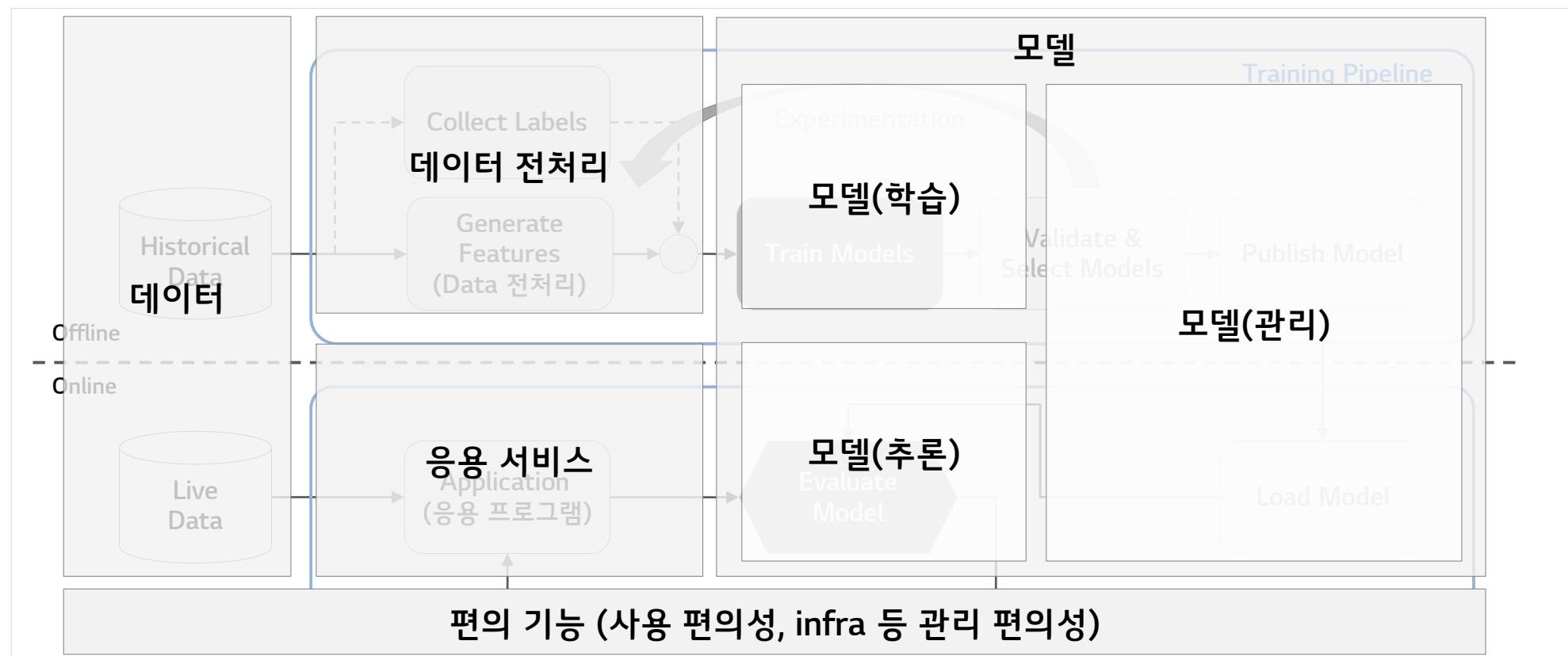
1) ML(Machine Learning), DL(Deep Learning) 포함

2) 출처: AI ML/DL Pipeline - Netflix, Inc.

I. 딥러닝 기술 Stack

- 딥러닝 Pipeline은 데이터, 데이터 전처리, 모델, 응용서비스와 편의 기능으로 구분됨

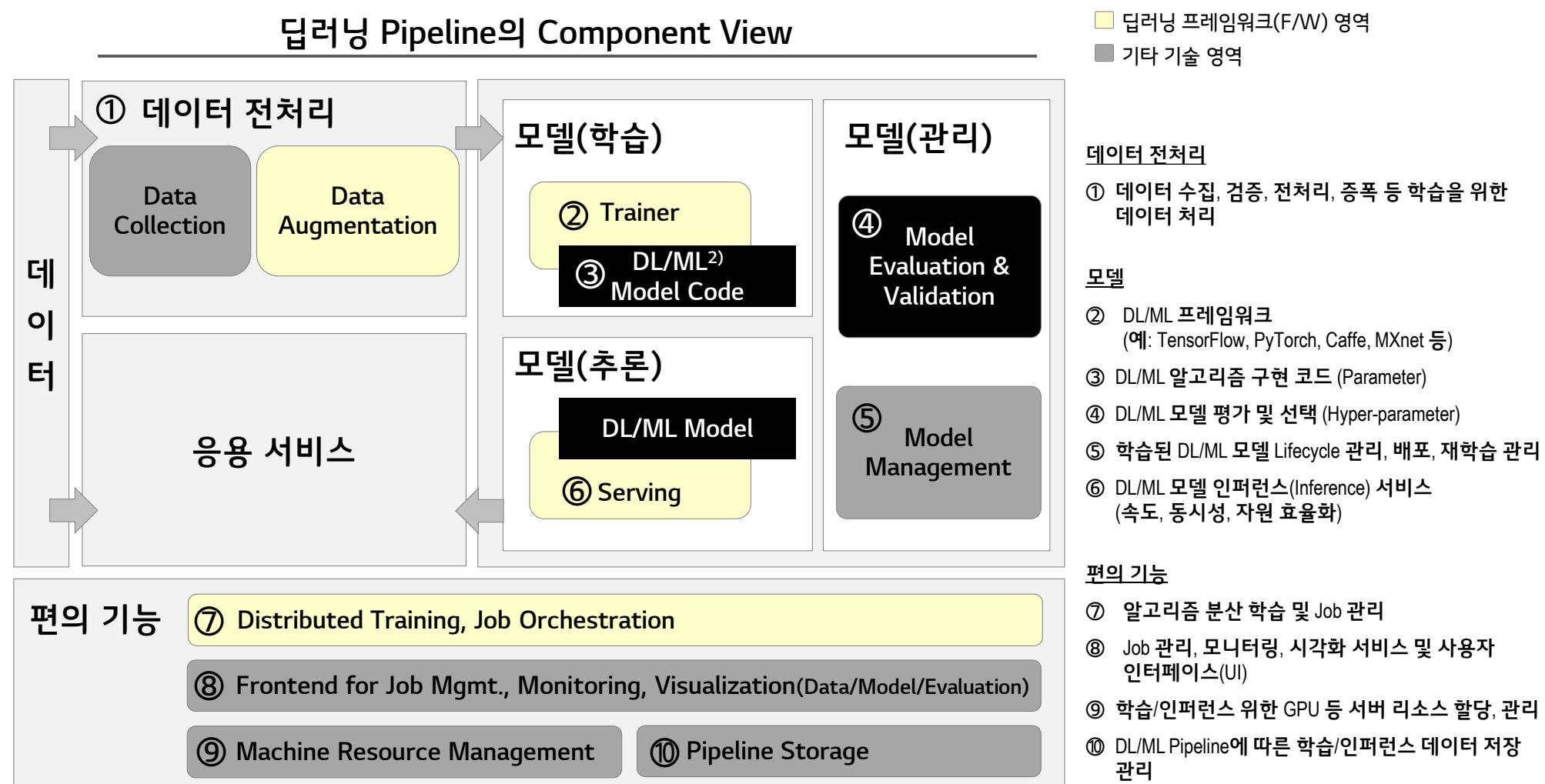
딥러닝 Pipeline의 Logical View



1) 출처: Hidden Technical Debt in Machine Learning Systems (NIPS 2015)

I. 딥러닝 기술 Stack

- Component View로 상세화 시 딥러닝 알고리즘, 딥러닝 프레임워크, 기타 기술 영역으로 구분됨



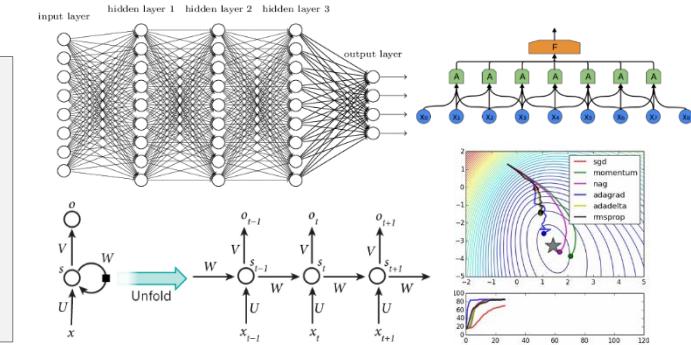
1) 참고: Hidden Technical Debt in Machine Learning Systems (NIPS 2015), TFX: A TensorFlow-Based Production-Scale Machine Learning Platform (KDD 2017)

2) DL: Deep Learning, ML: Machine Learning

I. 딥러닝 기술 Stack

- 딥러닝 기술 Stack 중 소프트웨어 영역은 알고리즘과 프레임워크로 나누어 볼 수 있음

✓ 딥러닝 알고리즘



✓ 딥러닝 프레임워크(F/W)



데이터

H/W
(GPU, AI Chipset ...)





딥러닝 알고리즘 동향



II. 딥러닝 알고리즘 동향

■ AlphaGo의 진화

| 버전 | Elo Rating | 경기 결과 | 특징 |
|-------------|------------|-------------------------|---|
| AlphaGo Fan | 3144 | • 5:0 승 (vs. 판후이 프도 2단) | <ul style="list-style-type: none">• 48가지¹⁾ 특징(feature) 정보 입력• 사람의 기보(16만개) 학습 중심²⁾ |
| AlphaGo Lee | 3739 | • 4:1 승 (vs. 이세돌) | <ul style="list-style-type: none">• AlphaGo Fan 개선• 더 큰 인공신경망(Deep Neural Nets) 사용 |

1) 돌 색상, 단수, 측 등의 정보

2) 정확하게는 사람의 기보 학습 후 정책망 Self-Play 기보로 가치망을 학습한다.

II. 딥러닝 알고리즘 동향

■ AlphaGo의 진화

| 버전 | Elo Rating | 경기 결과 | 특징 |
|--------------------------------------|------------|--|---|
| AlphaGo Fan | 3144 | • 5:0 승 (vs. 판후이 프도 2단) | <ul style="list-style-type: none"> 48가지¹⁾ 특징(feature) 정보 입력 사람의 기보(16만개) 학습 중심²⁾ |
| AlphaGo Lee | 3739 | • 4:1 승 (vs. 이세돌) | <ul style="list-style-type: none"> AlphaGo Fan 개선 더 큰 인공신경망(Deep Neural Nets) 사용 |
| '17.05 AlphaGo Master | 4858 | <ul style="list-style-type: none"> 60:0 승 (vs. 타이젬 프도선수들) 3:0 승 (vs. 커제) | <ul style="list-style-type: none"> AlphaGo Lee 개선 자가 대국(Self-Play) 기보 학습(강화학습) 중심 <ul style="list-style-type: none"> ※ 사람의 기보 학습 후 초기값으로 사용 최신 인공신경망(Residual Nets) 사용 및 정책망, 가치망 동합 |
| '17.10 AlphaGo Zero ³⁾ | 5185 | <ul style="list-style-type: none"> 100:0 승 (vs. AlphaGo Lee) 89:11 승 (vs. AlphaGo Master) | <ul style="list-style-type: none"> 2가지(돌 색상: 흑/백) 특징 정보만 입력 자가 대국(Self-Play) 기보만(강화학습) 학습 <ul style="list-style-type: none"> ※ 사람의 기보를 전혀 사용하지 않음 최신 인공신경망(Residual Nets) 사용 및 정책망, 가치망 동합 |
| '17.12 AlphaZero ⁴⁾ | 4000(추정) | <ul style="list-style-type: none"> 60:40 승 (vs. 3일 AlphaGo Zero) 28:0 승 (vs. Stockfish - 체스) 90:8 승 (vs. Elmo - 일본식 장기) | <ul style="list-style-type: none"> AlphaGo Zero 범용화 (바둑, 체스, 일본식 장기) 바둑, 체스, 일본식 장기 규칙만 입력 후 자가 대국으로 학습 |

1) 돌 색상, 단수, 측 등의 정보

3) 40일 학습

2) 정확하게는 사람의 기보 학습 후 정책망 Self-Play 기보로 가치망을 학습한다.

4) 34시간 학습

II. 딥러닝 알고리즘 동향

▪ AlphaGo의 진화 시사점

- 범용적 AI 가능성 시사
 - 강화학습(Reinforcement Learning)의 높은 성능 증명
 - AlphaGo Zero 접근법 성공에 대한 논리적 근거 부족 (인공신경망 인과관계 설명 난해)
- 설명 가능 인공지능(Explainable AI) 중요성 부각

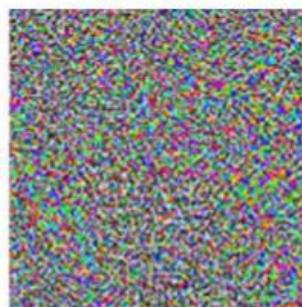
II. 딥러닝 알고리즘 동향

▪ AI 보급, 확산에 따른 신뢰성 요구 증가

- 특정 노이즈에 오인식 발생



$+ \epsilon$

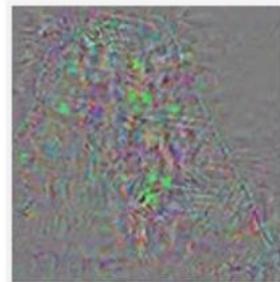


=



“판다” (57.7% 확신)

“인셉션 V3” (99.3% 확신)



“사원” (97% 확신)

“인셉션 V3” (98% 확신)

※ 컨테이너의 옆 면을 하늘로 착각한 자율주행차



II. 딥러닝 알고리즘 동향

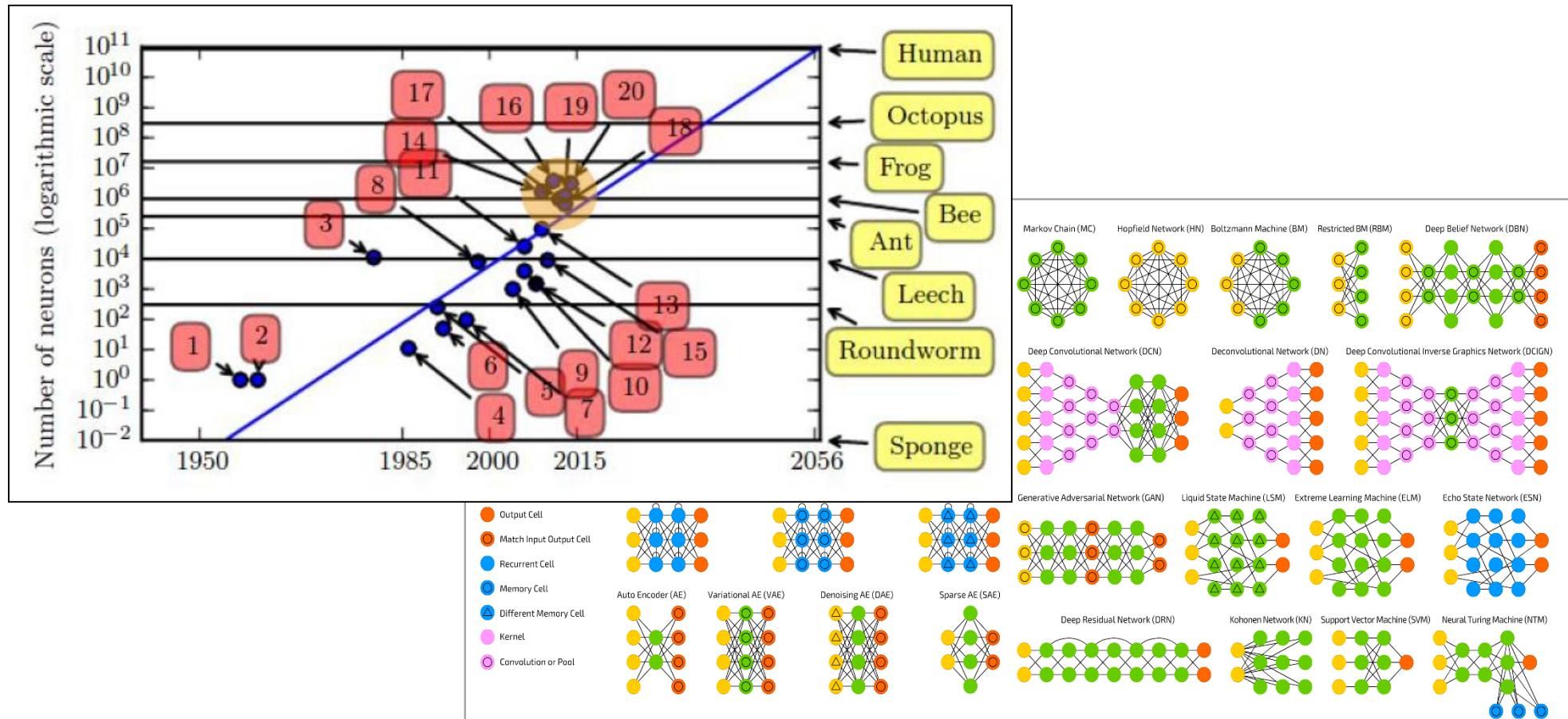
▪ 신뢰성 있는 AI 요구 시사점

- 적대적 공격 등의 노이즈에 강건한 신뢰성 있는 딥러닝 모델 중요
- AI, 딥러닝이 판단한 근거, 판단 과정의 타당성 설명 필요
→ 설명 가능 인공지능(Explainable AI)

II. 딥러닝 알고리즘 동향

▪ 높은 정확도를 위해 인공신경망 규모 증가

- 신경망 깊이가 늘어남(예, 152층), 학습할 파라미터(Parameter) 증가
- 다양한 구조의 신경망 등장



II. 딥러닝 알고리즘 동향

▪ 인공신경망 규모 증가 시사점

- 인공신경망 및 알고리즘 복잡성 증가에 따른 자동화 필요
 - 수 많은 하이퍼파라미터(hyper-parameter) 선택
 - 복잡한 신경망 구조 고안 및 최적화
- 인공신경망 경량화 필요
 - 판정 속도 개선
 - 메모리 절약으로 임베디드 등에 적용 가능

II. 딥러닝 알고리즘 동향

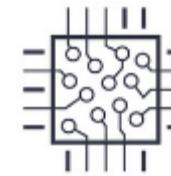
- 딥러닝 알고리즘 시사점 요약



범용적 AI



강화학습



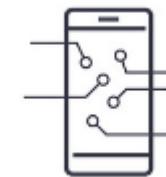
강건한 딥러닝 모델



설명 가능 AI



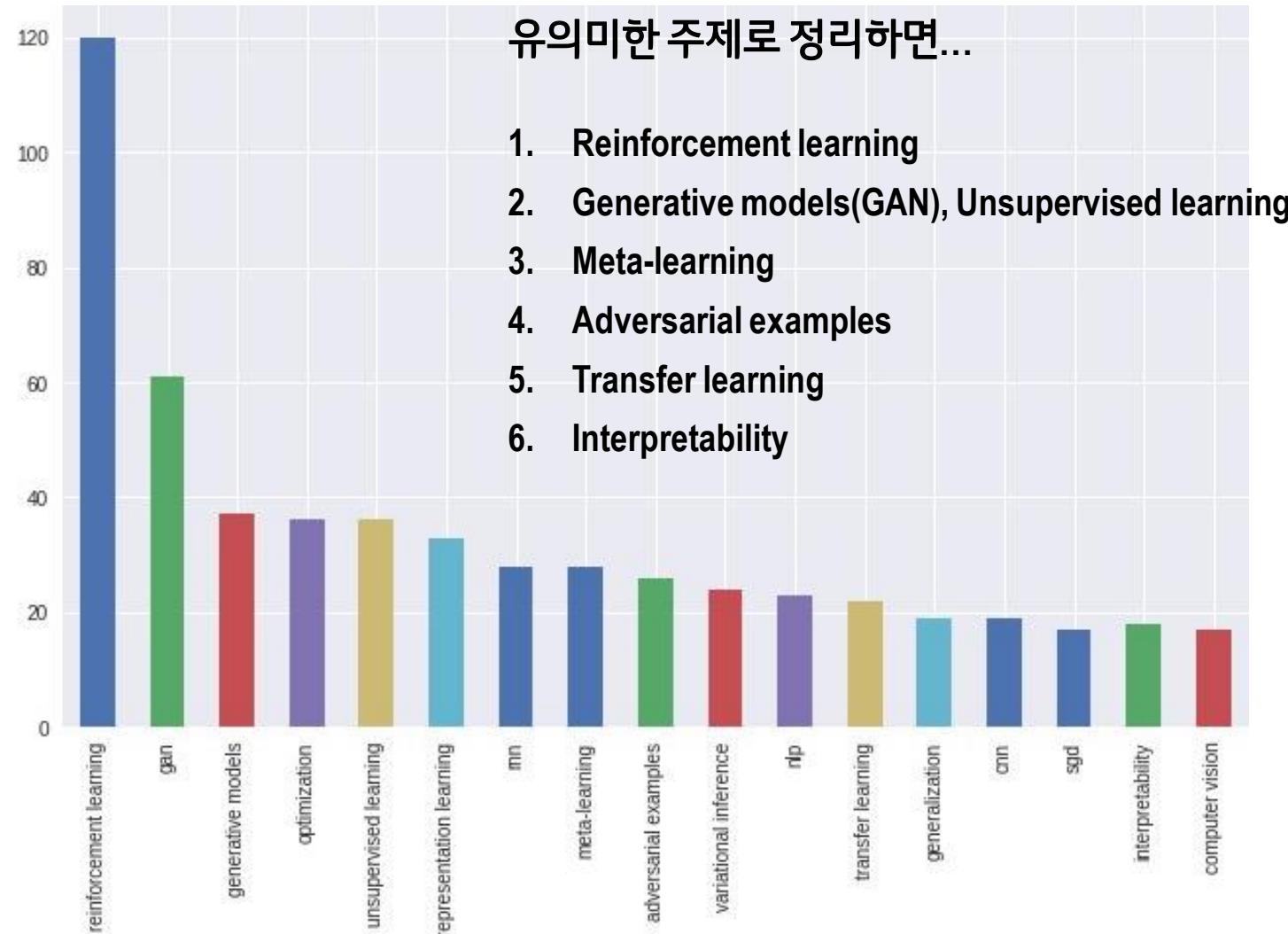
딥러닝 자동화



딥러닝 모델 경량화

II. 딥러닝 알고리즘 동향

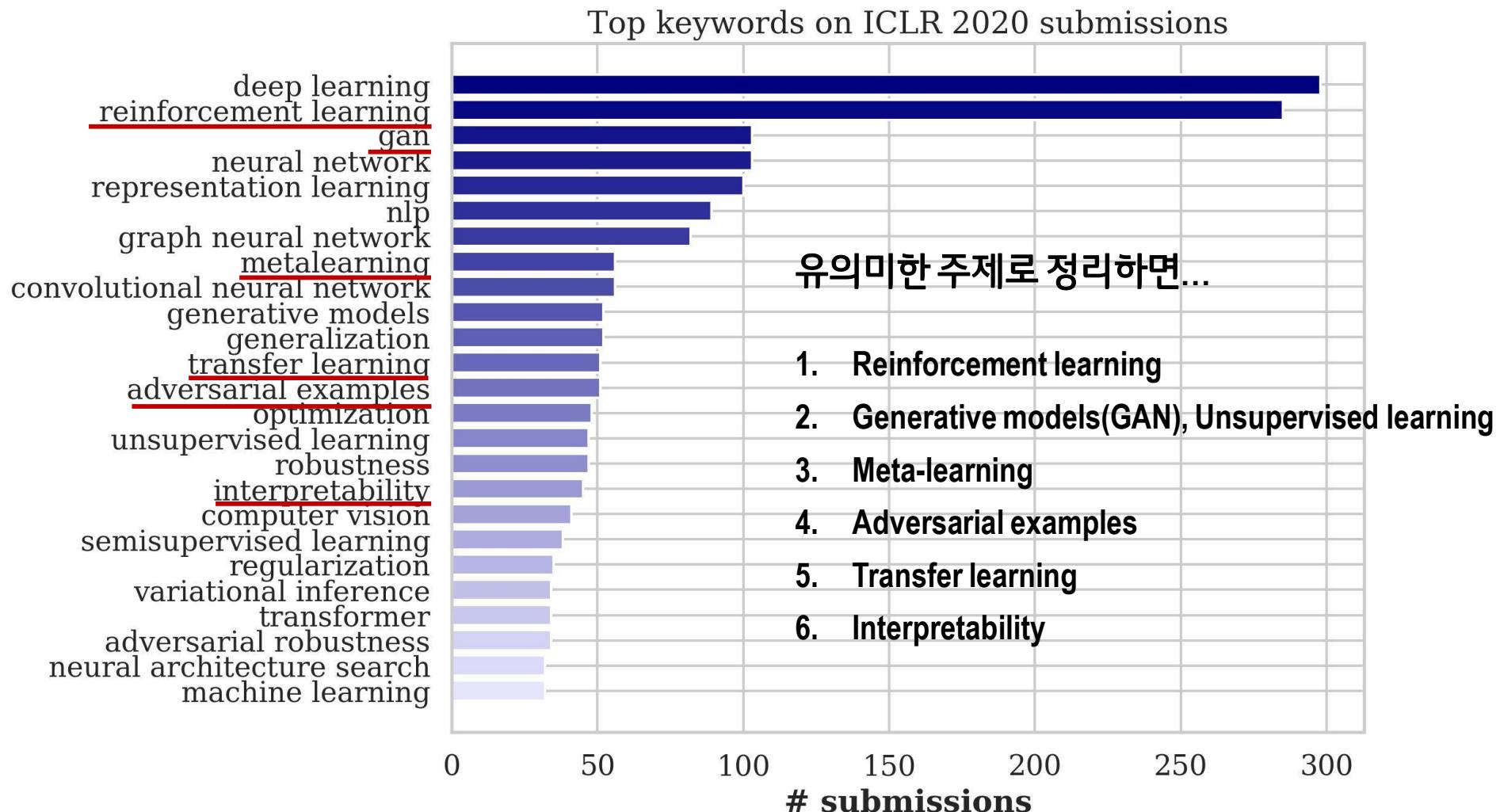
■ ICLR¹⁾ 2019 논문 주제 Top-20 정리



1) ICLR : International Conference for Learning Representations, 2013년 부터 시작된 딥러닝 전문 국제 학술대회, <https://iclr.cc/>

II. 딥러닝 알고리즘 동향

■ ICLR¹⁾ 2020 논문 주제 Top-25 정리



1) ICLR : International Conference for Learning Representations, 2013년 부터 시작된 딥러닝 전문 국제 학술대회, <https://iclr.cc/>

II. 딥러닝 알고리즘 동향

▪ 딥러닝 알고리즘 동향 정리



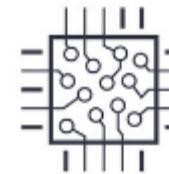
범용적 AI

Transfer learning
(Domain adaptation)



강화학습

Reinforcement learning



강건한 딥러닝 모델

Adversarial examples



설명 가능 AI

Interpretability



딥러닝 자동화

Meta-learning



생성 모델

Generative models(GAN)

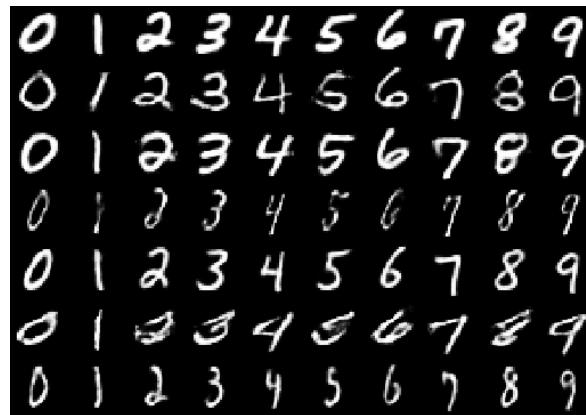


딥러닝 모델 경량화

II. 딥러닝 알고리즘 동향

- Domain Adaptation: Source Domain 데이터 학습만으로도 Target Domain에 사용

✓ 필기체 숫자(Source Domain) 학습

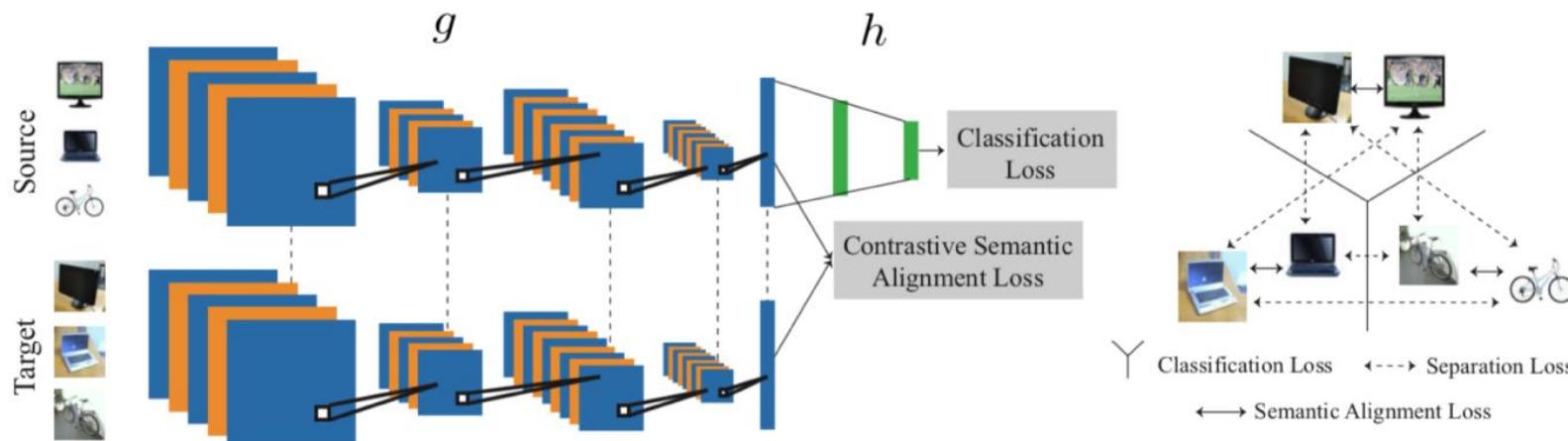


학습된 딥러닝 모델로
Target Domain 판정

✓ 간판 번호(Target Domain) 판정



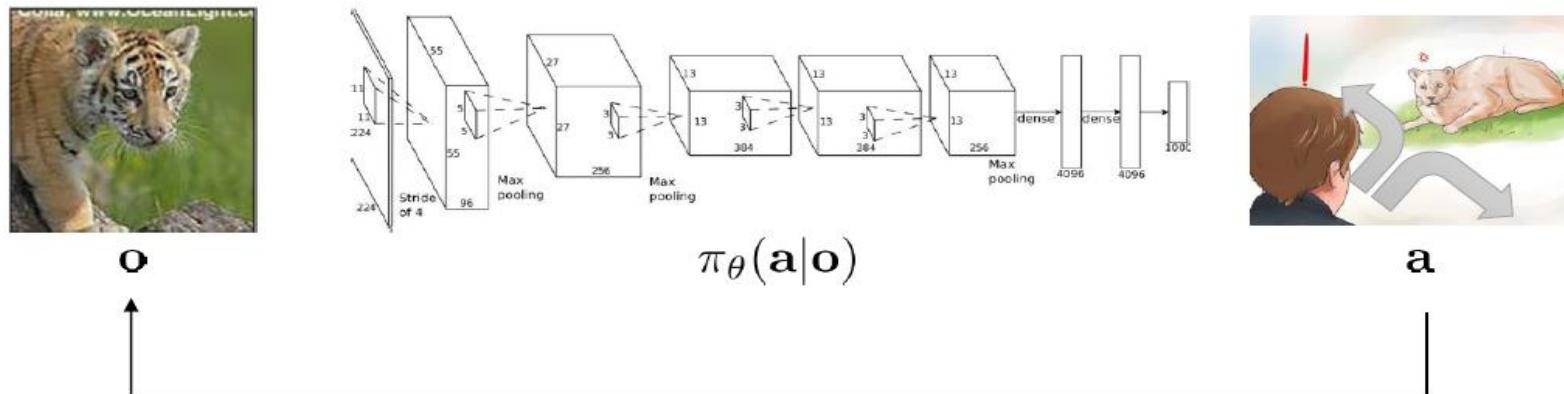
- Source Domain 데이터 학습과 동시에 Target Domain Align, Separation 정보도 학습



II. 딥러닝 알고리즘 동향

■ 심층 강화학습 (Deep Reinforcement Learning)

- Deep neural network + 강화 학습 = Deep reinforcement learning (e.g. Deep Q-Network)



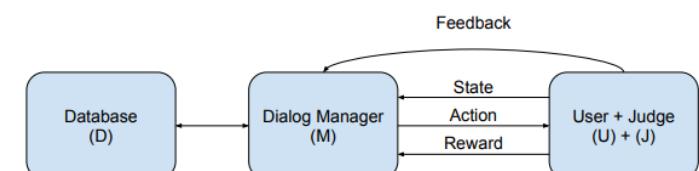
✓ Decision Making 및 Control 문제 해결에 적합

- 단순 분류(Classification)가 아닌 연속적인 의사결정 및 그에 따른 실행 판단
(예) 로보틱스(Robotics), 자율주행차, 게임 에이전트, 시뮬레이션 최적화 사례



✓ 기존 딥러닝 알고리즘 개선에 활용

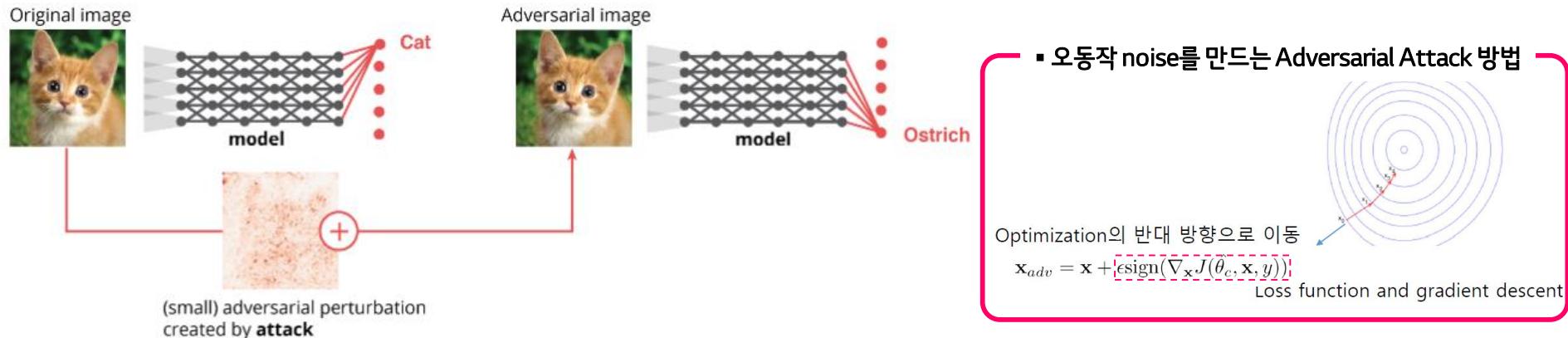
- 지도학습으로 학습된 인공신경망 모델 정확도를 개선하기 위해 추가적으로 강화학습 적용
(예) AutoML, 대화 에이전트(챗봇) 정확도 개선 사례



II. 딥러닝 알고리즘 동향

▪ Robust Model(Adversarial Defender)

✓ 현재 딥러닝 모델(ConvNet, RNN)은 작은 noise에 쉽게 오동작할 수 있음: Adversarial Attack \leftrightarrow Adversarial Defender



• Adversarial Training + Denoiser로 Robust Model 진화, 발전 중

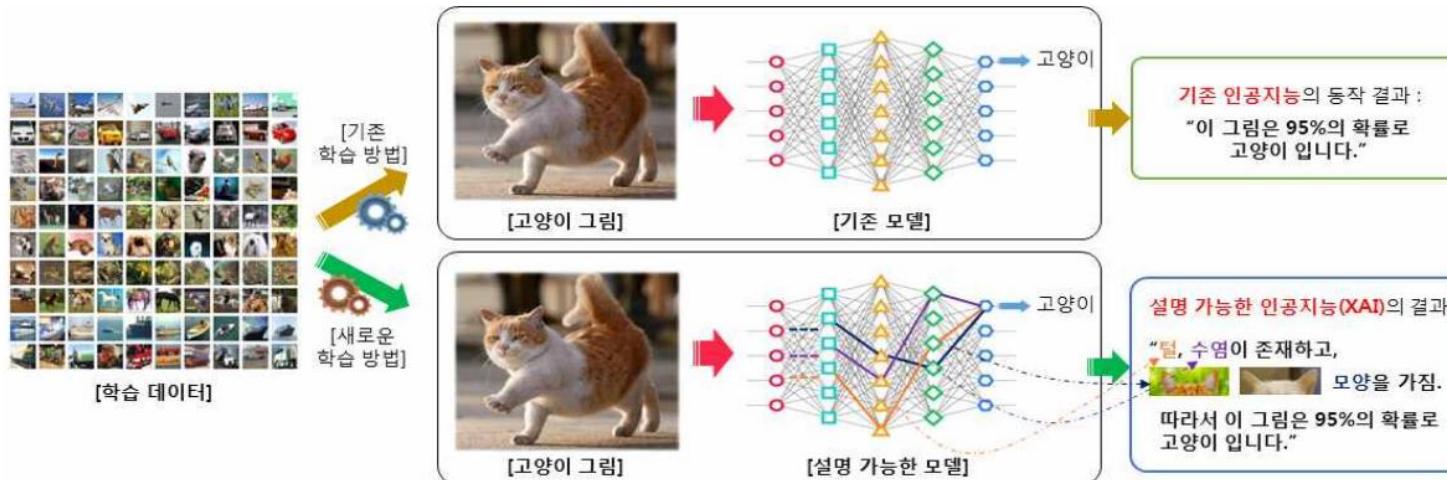
- 오동작 noise를 주입하여 학습(i.e. 백신), 또는 noise 패턴 검출 학습하여 사전 차단
- Denoiser를 학습시켜 오동작 noise를 줄임



II. 딥러닝 알고리즘 동향

▪ 설명 가능 AI (eXplainable AI, XAI)

- AI의 행위와 판단을 사람이 이해할 수 있는 형태로 설명할 수 있는 기술, 알고리즘 ►

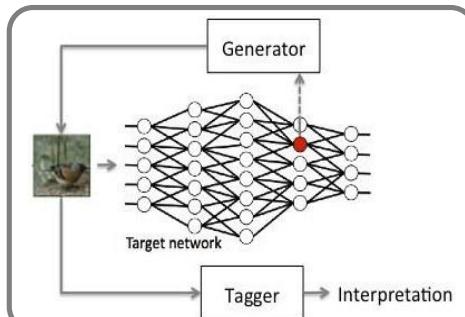


- XAI는 초기 기술 수준이며 현재 대표적인 몇 가지 구현 기술이 존재함

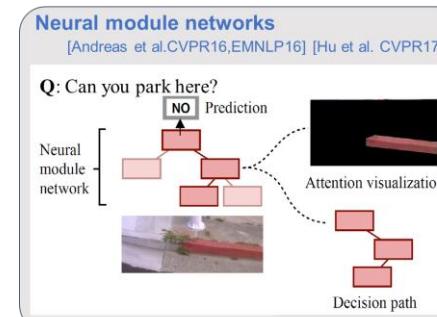
✓ Attention Mechanisms



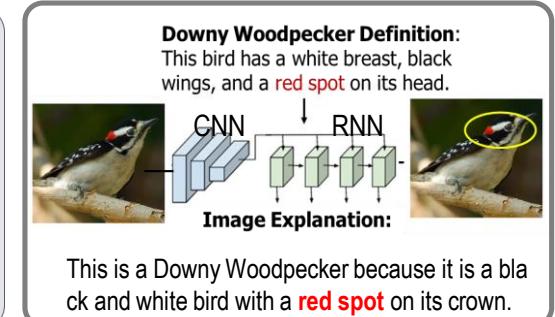
✓ Feature Identification



✓ Modular Networks



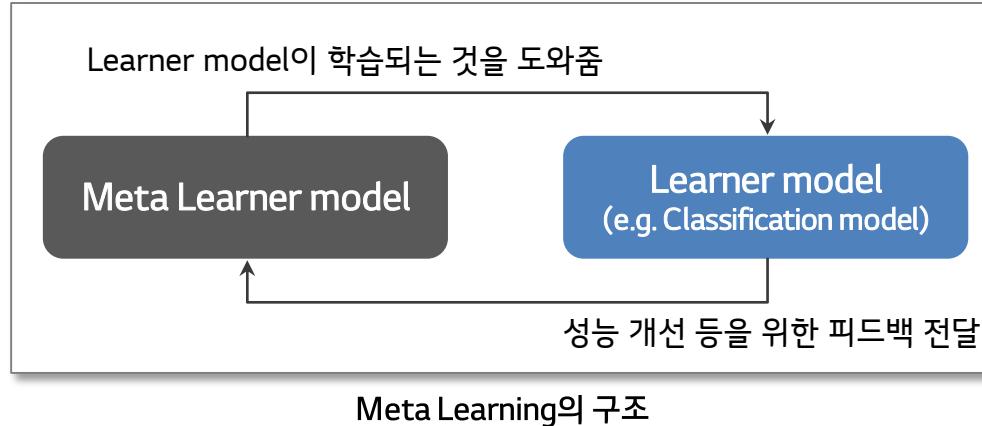
✓ Learn to Explain



II. 딥러닝 알고리즘 동향

▪ Meta Learning

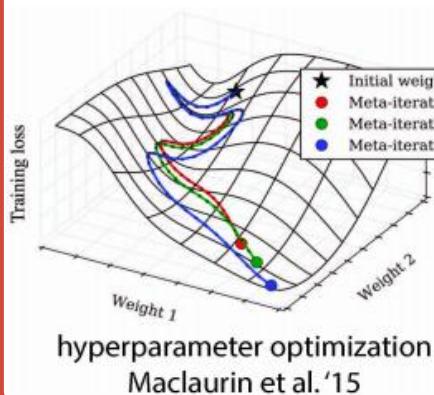
- 학습하는 방법을 학습 (Learning to learn) 하는 메타(Meta) 알고리즘의 총칭



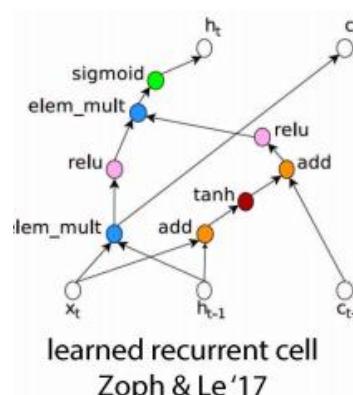
- ✓ 인공신경망 및 알고리즘 복잡도 증가 → 자동화
 - Automated ML (AutoML)
- ✓ 범용적 AI 지향 → 학습하는 방법을 알고 있는 AI
 - 소수의 학습 데이터 사용, 다양한 Task 수행 (Multi-Task)

- AutoML에 좋은 성능을 보이고 있으나 범용적 AI 태스크 수행은 아직 초기 단계임

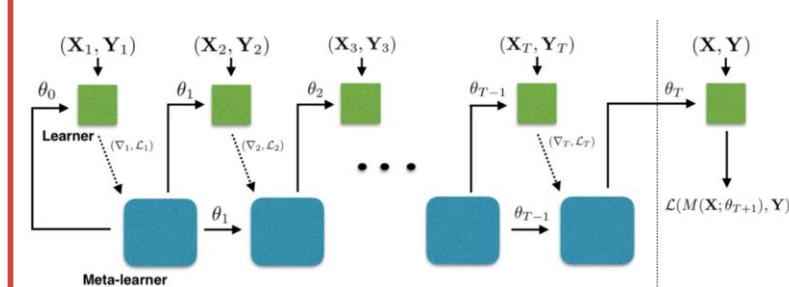
- ✓ 하이퍼파라미터 자동 최적화



- ✓ 최적의 신경망 모델 자동 탐색¹⁾



- ✓ 소수 데이터 학습 및 Unseen Task 수행

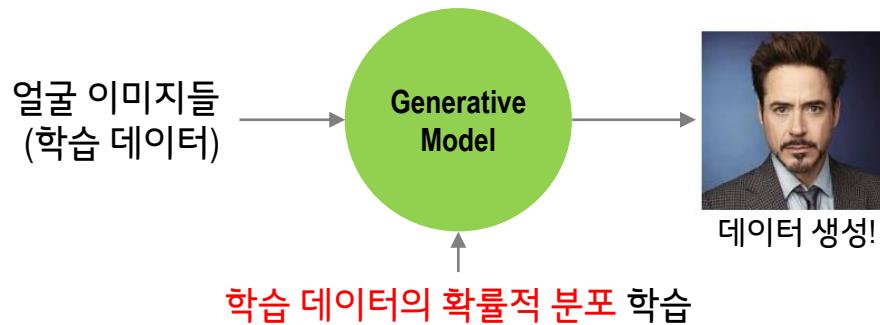


II. 딥러닝 알고리즘 동향

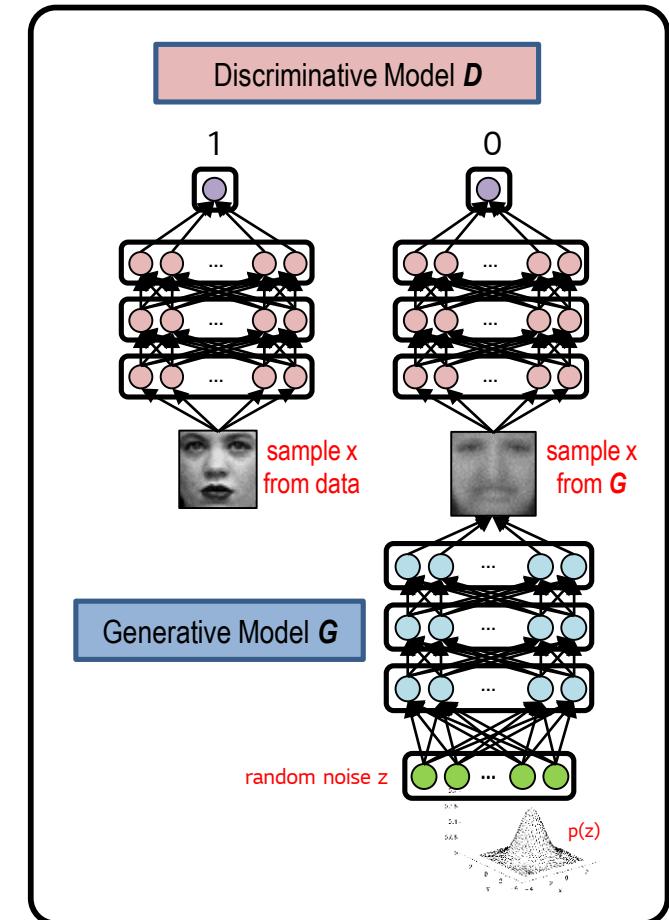
▪ Generative Model

- 기존 딥러닝 처럼 인식, 예측이 아니라 데이터 자체를 생성함. **Generative Adversarial Nets**이 대표적인 알고리즘

✓ Generative Model 개요



✓ GAN 작동 원리



- GAN은 초기 기술이며 학습 데이터 확보 목적으로 활용

✓ 하나의 동영상으로 스타일, 포즈, 배경을 변경하여 생성

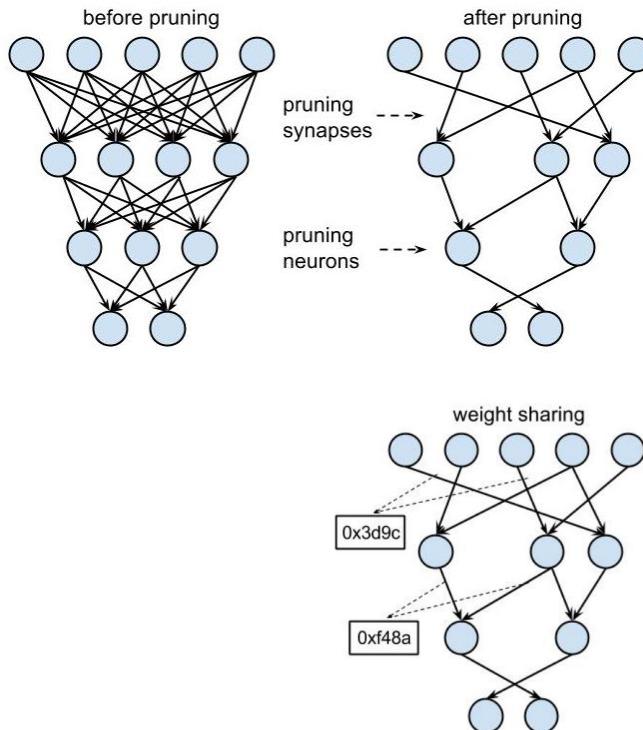


II. 딥러닝 알고리즘 동향

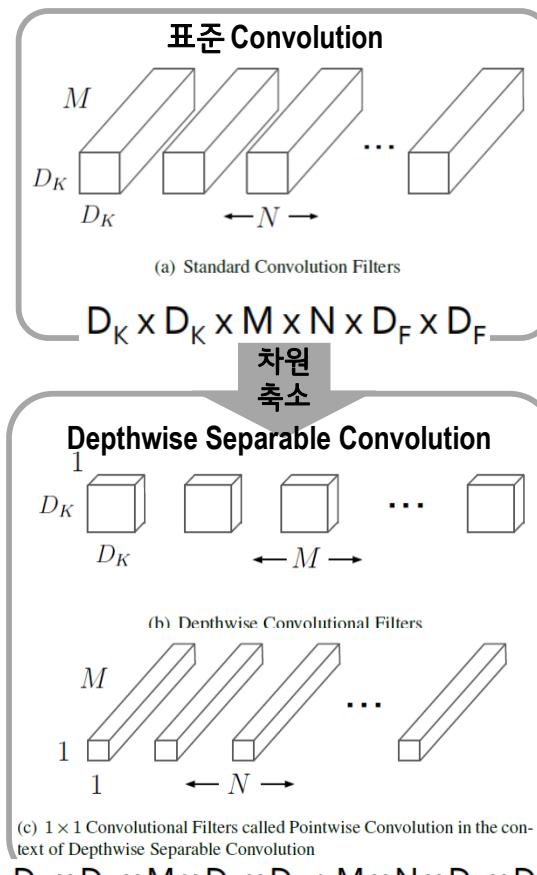
▪ 딥러닝 모델 경량화(압축)

- 정확도 손실 없이 인공신경망의 크기(parameter 개수)를 줄여 속도를 높이고 메모리 사용량을 줄임

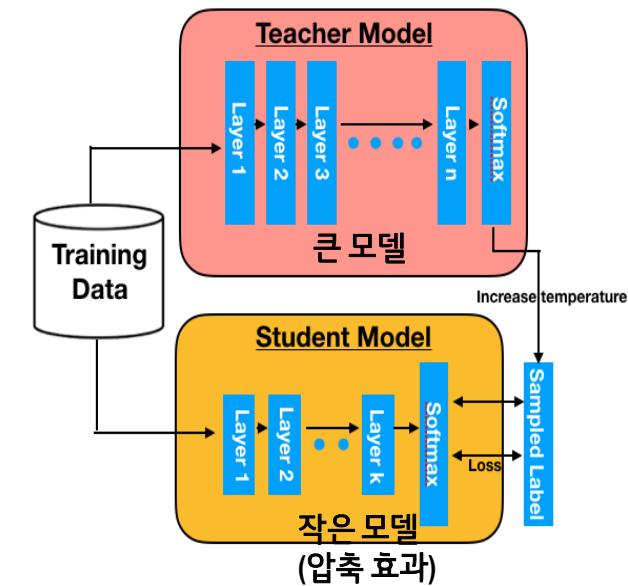
✓ 파라미터 가지치기(Pruning) 및 공유



✓ 차원 축소(reduction)



✓ Knowledge distillation (지식 증류)



축소 효과: $1/N + 1/D_K^2$

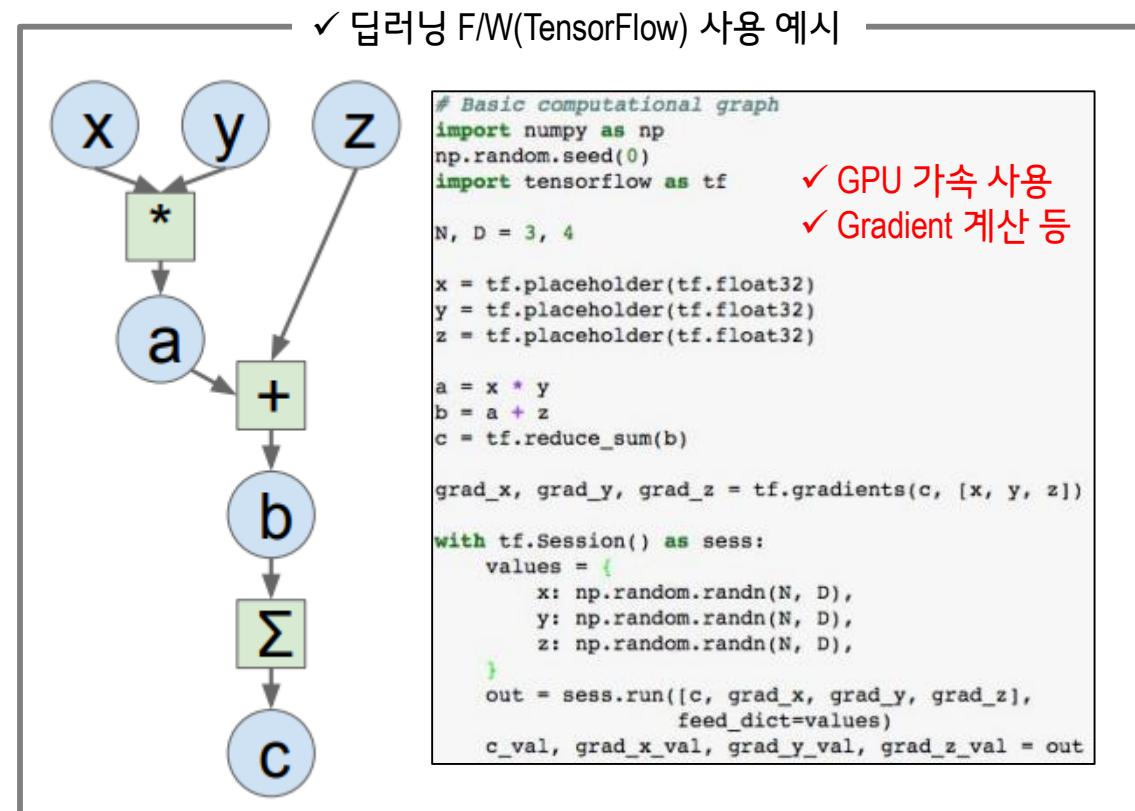


딥러닝 F/W 기술 동향

III. 딥러닝 F/W 기술 동향

▪ 딥러닝 F/W(Framework) 정의 및 역할

- GPU 가속을 활용하여 대규모 행렬 고속 계산(linear algebra), 수학 알고리즘 고속 수행
- 딥러닝 알고리즘 개발, 학습, 실행에 필요한 연산(gradient, exp 등) 및 함수(conv, relu 등) 제공
- 딥러닝 모델 재사용 (Fine-Tuning, Transfer Learning 등)

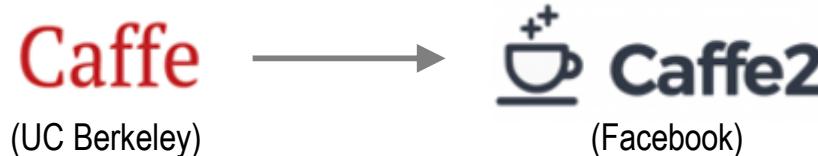
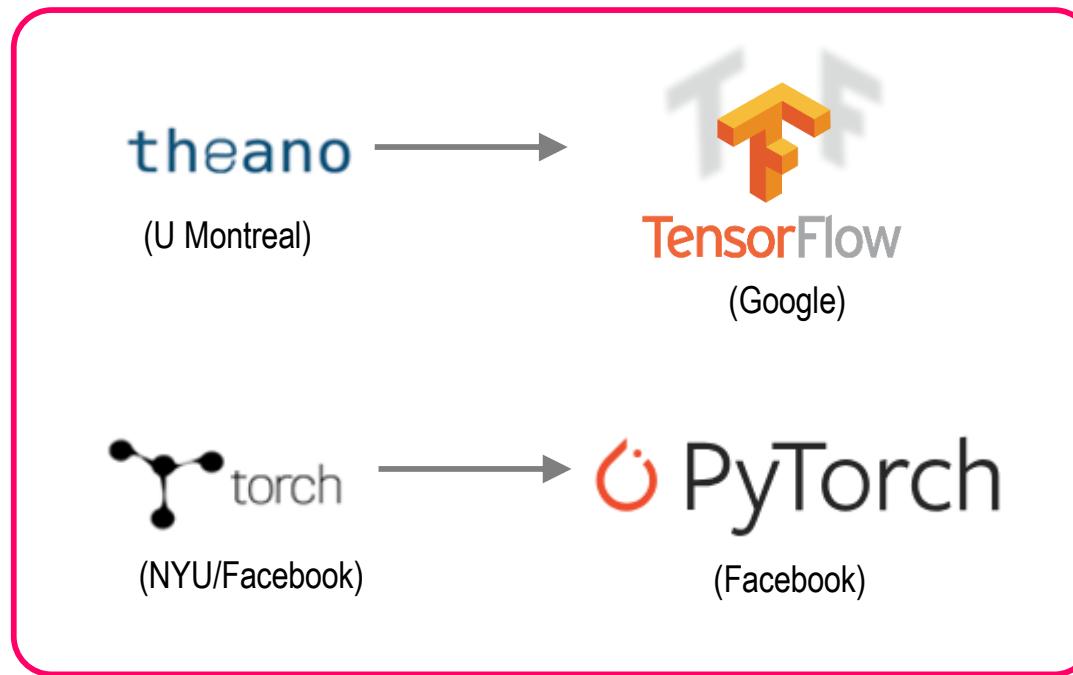


1) General-Purpose computing on Graphics Processing Units, GPU에서 응용 프로그램 계산 수행

III. 딥러닝 F/W 기술 동향

- 딥러닝 F/W 종류¹⁾

- 약 20여개의 딥러닝 F/W 존재 (대부분 OSS²⁾ 라이선스)



Chainer
(Preferred Networks)



(Baidu)

....

Microsoft
CNTK
(Microsoft)

mxnet
(U Washington/Amazon)

1) <http://cs231n.stanford.edu/> 참조

2) Open Source Software

III. 딥러닝 F/W 기술 동향

▪ 딥러닝 F/W 비교



✓ Static Computational Graph (TF 1.x 기준)

- 그래프를 한번 만들어 여러 번 실행
- 최적화 및 직렬화 용이

✓ Framework 성숙도 높음

- 레퍼런스 구현 및 모델 다수
- 프레임워크로써의 편의 기능 제공
- 분산병렬 학습 기능 제공

실 서비스에 적합



✓ Dynamic Computational Graph ▶

- 매 Forward Pass마다 새로운 그래프 생성
- RNN 등 dynamic graph가 필요한 알고리즘 적합

✓ 개발 생산성 높음

- 개발, 디버깅 용이
- 직관적이고 깔끔한 API

알고리즘 프로토타이핑, 연구에 적합

1) <http://cs231n.stanford.edu/> 참조

III. 딥러닝 F/W 기술 동향

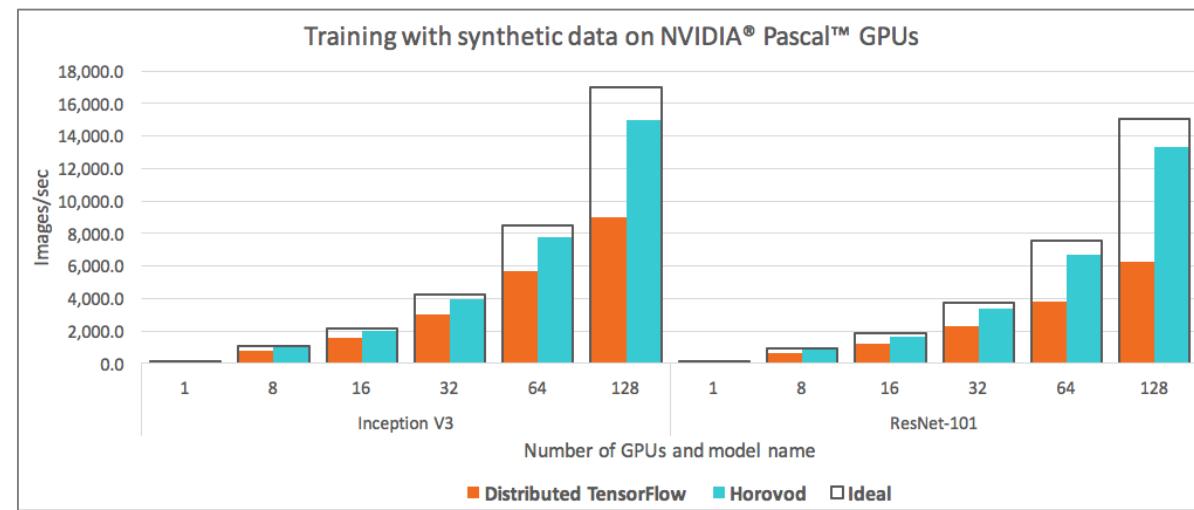
▪ 딥러닝 F/W 동향

- Dynamic Computational Graph 기능 제공으로 진화 (e.g. TensorFlow 2.0의 Eager Execution)
- 이종 F/W간 딥러닝 모델 호환 기술 등장: ONNX¹⁾



• 대규모 데이터 분산병렬 학습 딥러닝 F/W 진화: Horovod²⁾

- ✓ Data Parallelism, Model Parallelism 모두 지원
- ✓ TensorFlow, PyTorch 지원
- ✓ 분산 TensorFlow 보다 높은 성능
- ✓ MPI³⁾ 방식의 안정적인 분산처리 기능



1) <https://onnx.ai/> 2) <https://github.com/uber/horovod> 3) Message Passing Interface



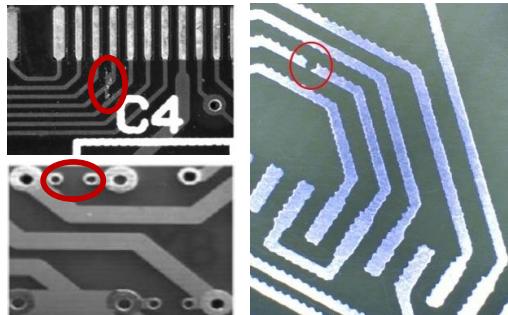
딥러닝 적용 사례 및 조언

IV. 딥러닝 적용 사례 및 조언

■ LG CNS의 딥러닝 사례

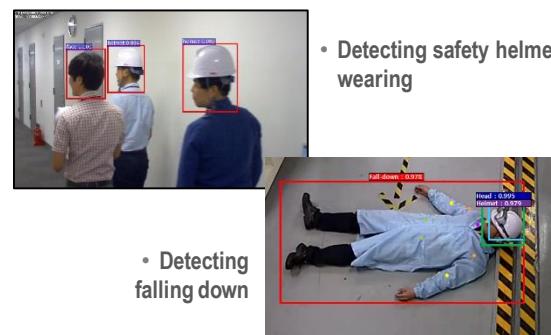
이미지 인식

제조 AI 비전검사



동영상 인식

AI CCTV 환경안전

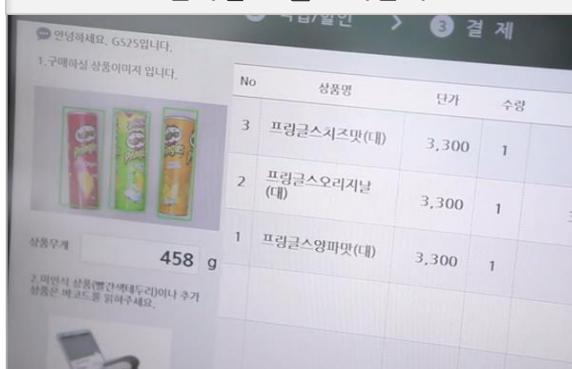


자연어 이해

텍스트봇 / 음성봇(스마트컨택센터)



편의점 AI 셀프 계산대



AI CCTV 보안



AI 튜터



IV. 딥러닝 적용 사례 및 조언

▪ 딥러닝 vs. 일반 소프트웨어

- AI/딥러닝 시스템과 일반적인 프로그램은 3가지의 차이점이 있음¹⁾

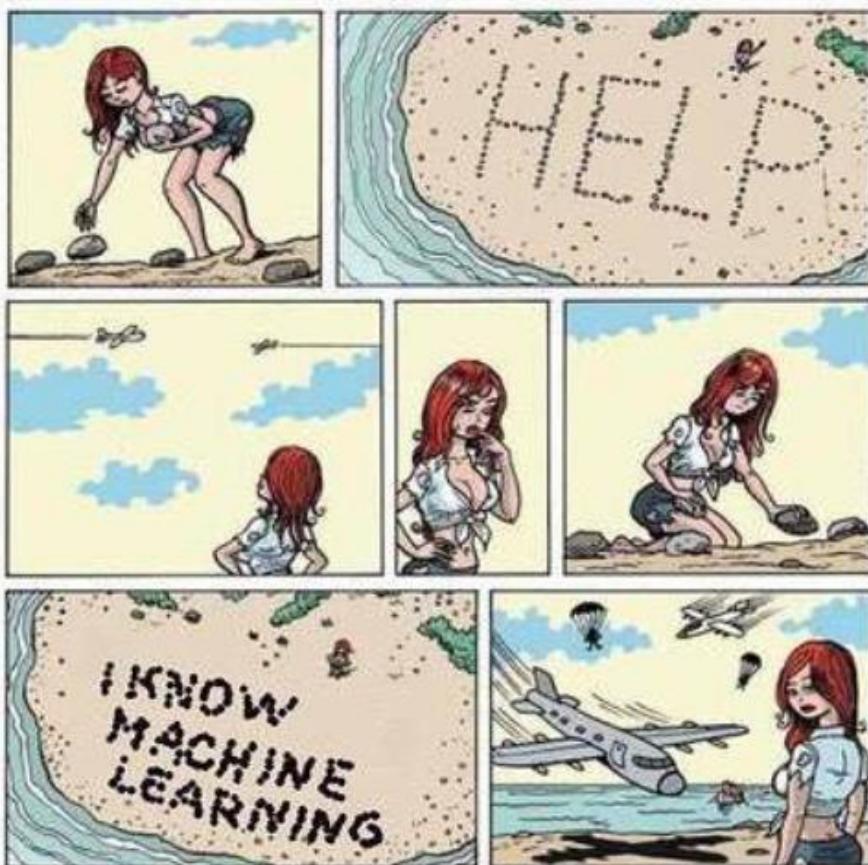
| AI/딥러닝 | 일반 소프트웨어 |
|---------------|--------------|
| 컨텐츠 | ▪ Models |
| 개발 과정 | ▪ Training |
| 수정/출시 | ▪ Retraining |
| ▪ Source Code | |
| ▪ Debugging | |
| ▪ Patching | |

AI/딥러닝은 기존 S/W 개발 방법과 다른 점을 이해할 필요가 있음.

1) Peter Norvig (Google 리서치 디렉터, 前 UC Berkeley 교수, AI: A Modern Approach 저자)

IV. 딥러닝 적용 사례 및 조언

▪ 인공지능, 빅데이터 전공 후 졸업 경로



AI job titles with the highest salaries

| Rank | Job title | Average salary |
|------|---------------------------|----------------|
| 1. | Machine learning engineer | \$142,858.57 |
| 2. | Data scientist | \$126,927.41 |
| 3. | Computer vision engineer | \$126,399.81 |
| 4. | Data warehouse architect | \$126,008.25 |
| 5. | Algorithm engineer | \$109,313.51 |



Source: Indeed

indeed

출처: indeed.com, 2019

IV. 딥러닝 적용 사례 및 조언

■ 머신러닝(딥러닝) 엔지니어의 역할 :

- Develop machine learning models, Improve existing machine learning models
- Implement machine learning algorithms and libraries
- Be in charge of the entire lifecycle (research, design, experimentation, development, deployment, monitoring, and maintenance)
- Collaborate with data engineers to develop data and model pipelines
- Apply machine learning and data science techniques and design distributed systems
- Write production-level code, Bring code to production
- Engage in code reviews
- Produce project outcomes and isolate issues
- Communicate complex processes to business leaders
- Analyze large and complex data sets to derive valuable insights
- Research and implement best practices to enhance existing machine learning infrastructure

출처: indeed.com

① 머신러닝(딥러닝) 알고리즘 역량

→ 단순 사용 수준이 아니라 원리를 알고 구현해 내는 수준



② SW 프로그래밍 및 시스템 개발 역량

→ 프로그래밍에 익숙하고 SW 개발 및 활용에 능숙한 수준

Q & A

감사합니다.

이주열 연구소장
jooyoul.lee@lgcns.com

별첨

별첨. 일반적 딥러닝 시스템 구성 요소

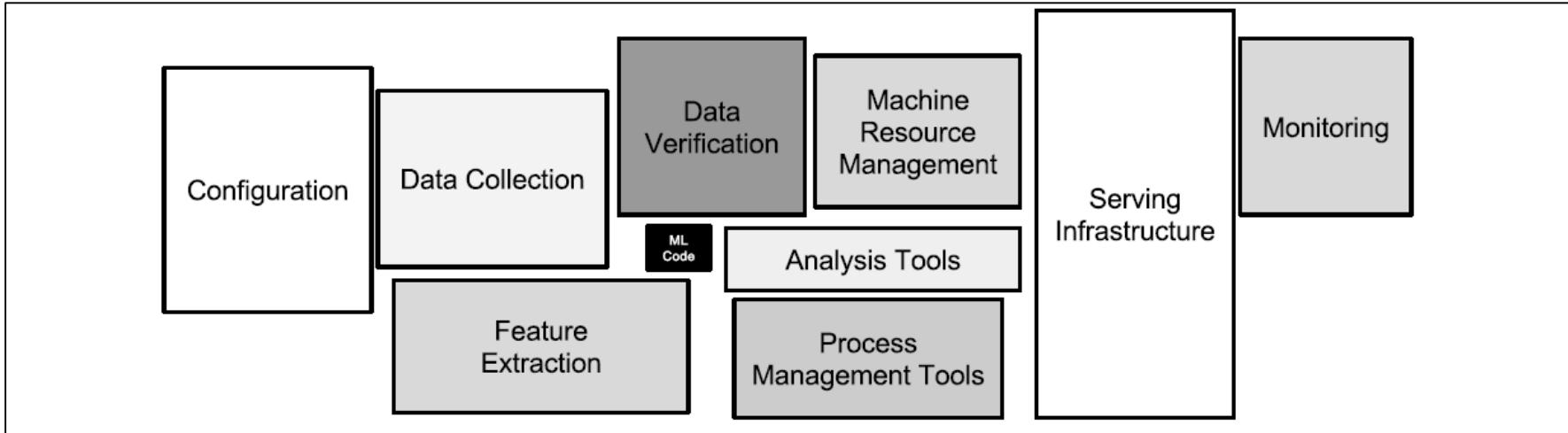


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

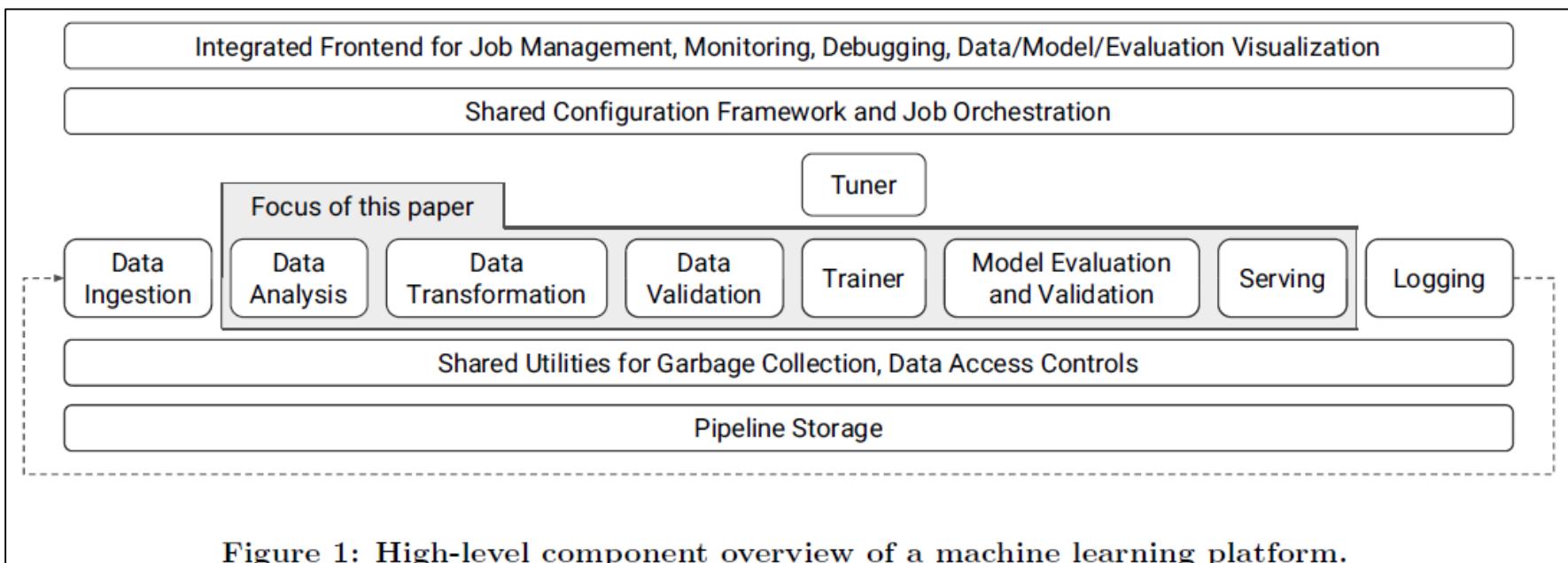
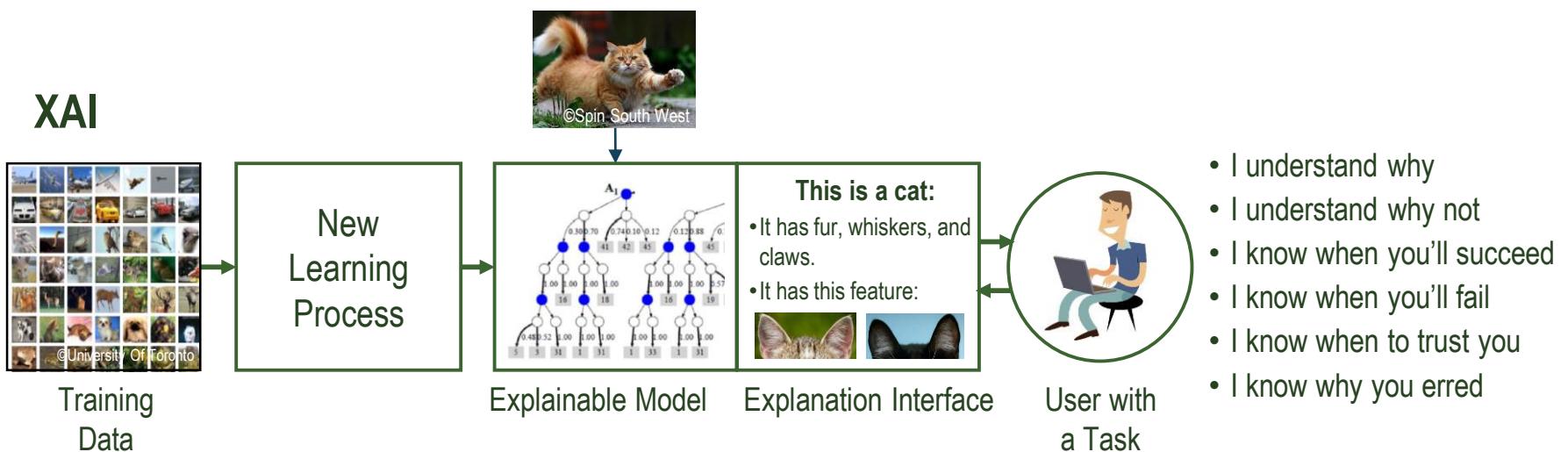
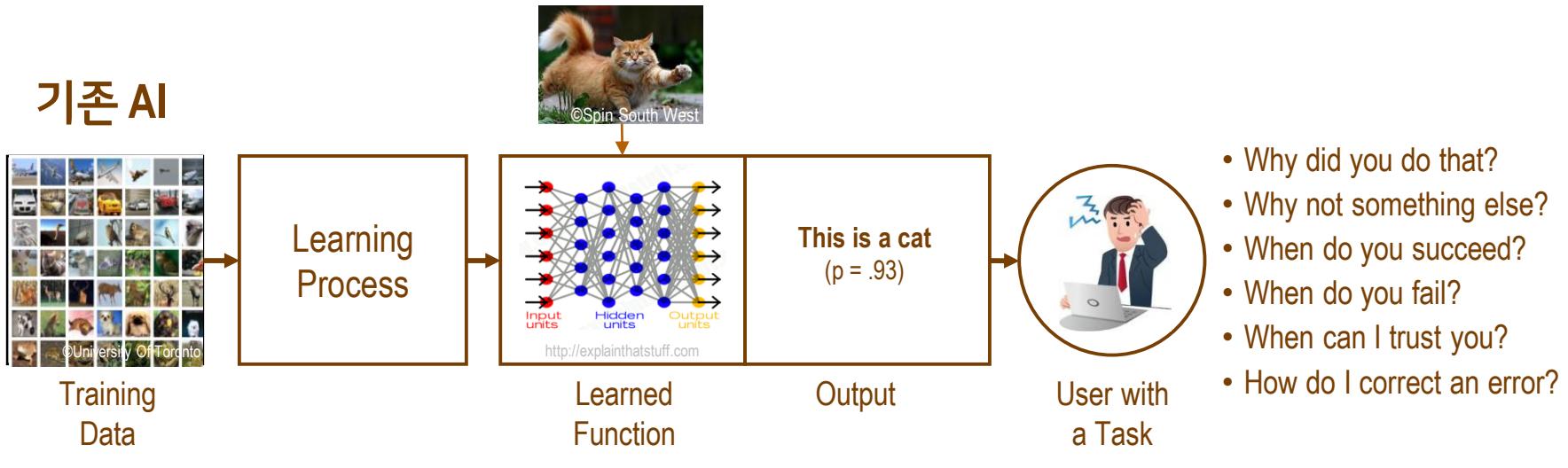


Figure 1: High-level component overview of a machine learning platform.



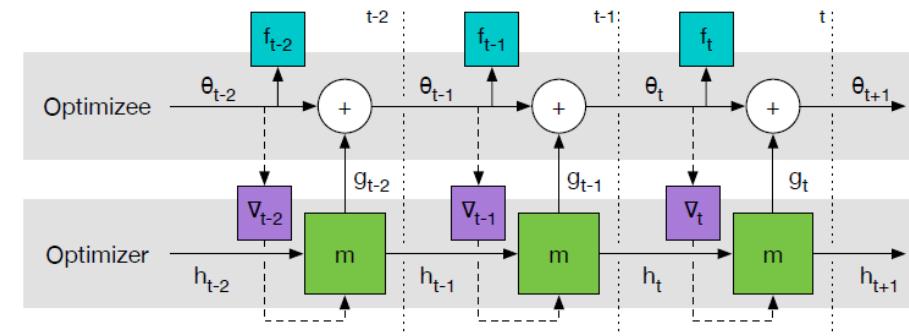
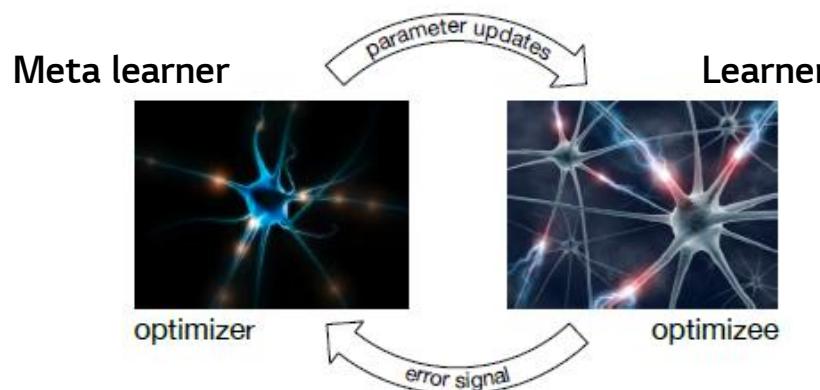
■ 기존 AI vs. XAI



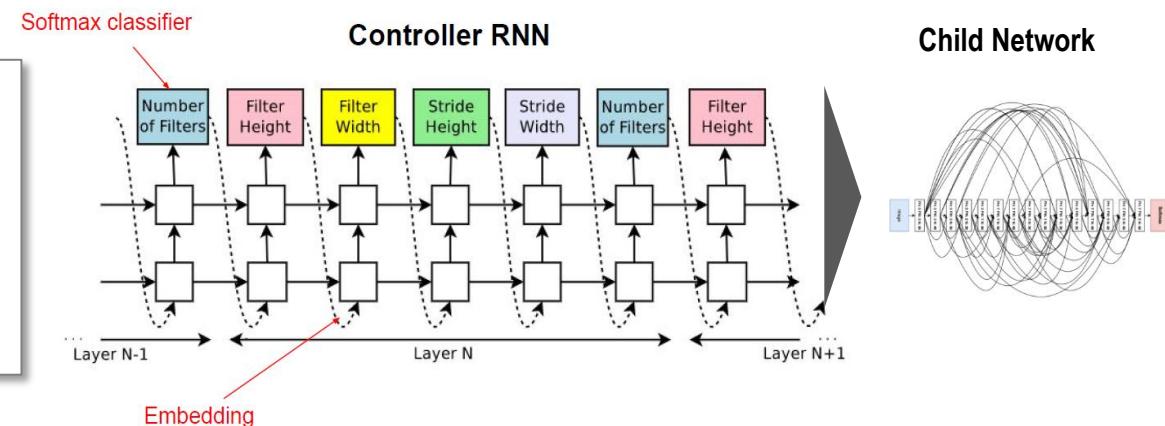
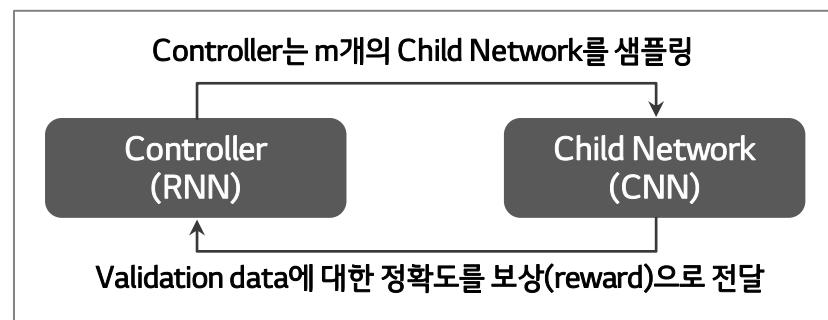
별첨. Meta Learning 작동 원리



- Meta learner는 주로 RNN¹⁾으로 구성하여 Controller 역할을 하고 Learner는 목적에 맞게 다양한 DNN 모델이 될 수 있음
 - ✓ 하이퍼파라미터 자동 최적화



- ✓ 최적의 인공신경망 모델 탐색 자동화



1) Recurrent Neural Nets

별첨. TensorFlow 1.x vs. PyTorch



TensorFlow: Build graph once, then run many times (**static**)

```
N, D, H = 64, 1000, 100
x = tf.placeholder(tf.float32, shape=(N, D))
y = tf.placeholder(tf.float32, shape=(N, D))
w1 = tf.Variable(tf.random_normal((D, H)))
w2 = tf.Variable(tf.random_normal((H, D)))

h = tf.maximum(tf.matmul(x, w1), 0)
y_pred = tf.matmul(h, w2)
diff = y_pred - y
loss = tf.reduce_mean(tf.reduce_sum(diff ** 2, axis=1))
grad_w1, grad_w2 = tf.gradients(loss, [w1, w2])

learning_rate = 1e-5
new_w1 = w1.assign(w1 - learning_rate * grad_w1)
new_w2 = w2.assign(w2 - learning_rate * grad_w2)
updates = tf.group(new_w1, new_w2)

with tf.Session() as sess:
    sess.run(tf.global_variables_initializer())
    values = {x: np.random.randn(N, D),
              y: np.random.randn(N, D),}
    losses = []
    for t in range(50):
        loss_val, _ = sess.run([loss, updates],
                             feed_dict=values)
```

Build
graph

PyTorch: Each forward pass defines a new graph (**dynamic**)

```
import torch

N, D_in, H, D_out = 64, 1000, 100, 10
x = torch.randn(N, D_in)
y = torch.randn(N, D_out)
w1 = torch.randn(D_in, H, requires_grad=True)
w2 = torch.randn(H, D_out, requires_grad=True)

learning_rate = 1e-6
for t in range(500):
    y_pred = x.mm(w1).clamp(min=0).mm(w2)
    loss = (y_pred - y).pow(2).sum()

    loss.backward()
```

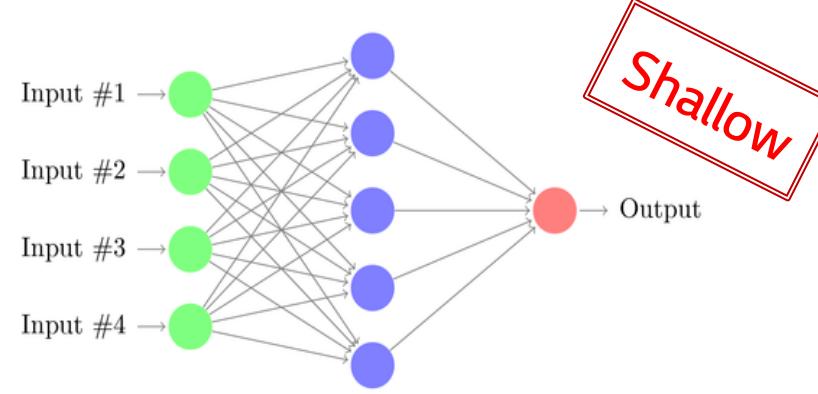
Run each
iteration

New graph each iteration

Deep Neural Networks + Machine Learning = Deep Learning

- Neural Networks

- 1~2개의 은닉층, 단순



- Deep Neural Networks

- 3개 이상의 은닉층, 복잡



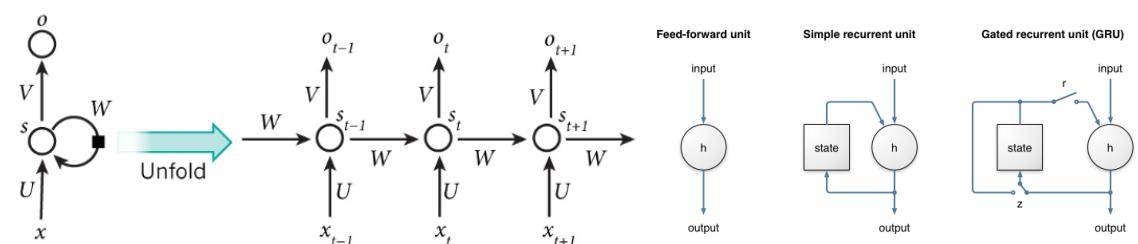
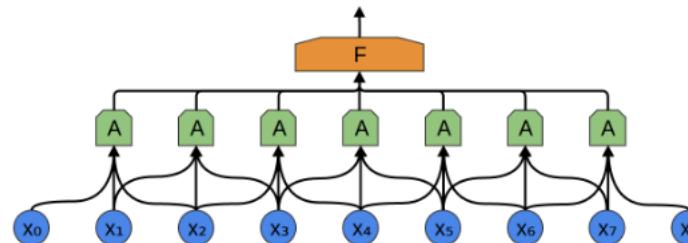
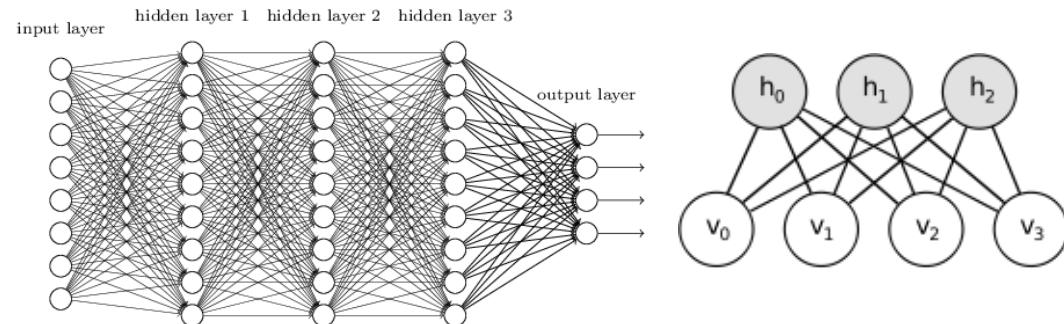
Ultra Deep!!!



Deep Neural Networks + Machine Learning = Deep Learning

다양한 구조의 깊은 인공신경망 존재

- Fully-Connected / Undirected
 - DNN(Deep Neural Networks)
 - RBM(Restricted Boltzmann Machine)
 - DBN(Deep Belief Network)
- Convolutional
 - LeNet
 - AlexNet, VGGNet
 - GoogleNet
 - ResNet
- Recurrent
 - LSTM(Long Short-Term Memory)
 - GRU(Gated Recurrent Unit)
 - SRU(Simple Recurrent Unit)



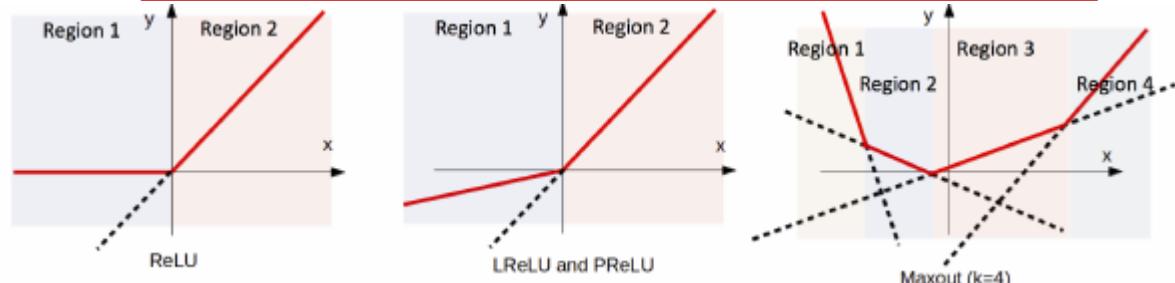


Deep Neural Networks + Machine Learning = Deep Learning

깊은 인공신경망을 Learning 시키는 다양한 기법 존재

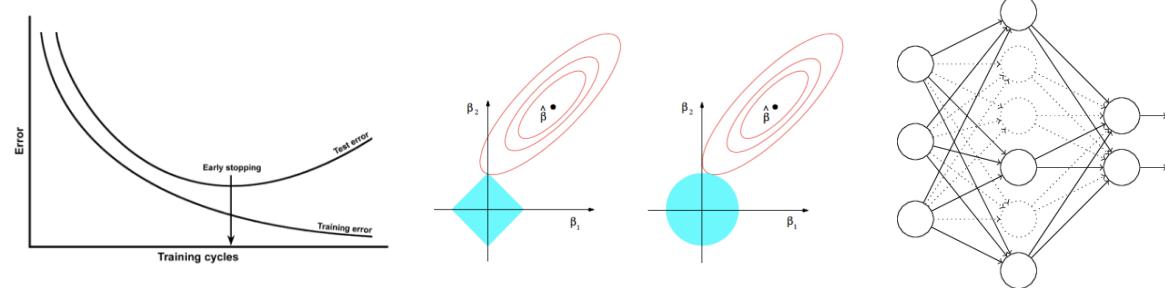
- Activation Function

- ReLU(Rectified Linear Units)
- Leaky ReLU
- Maxout



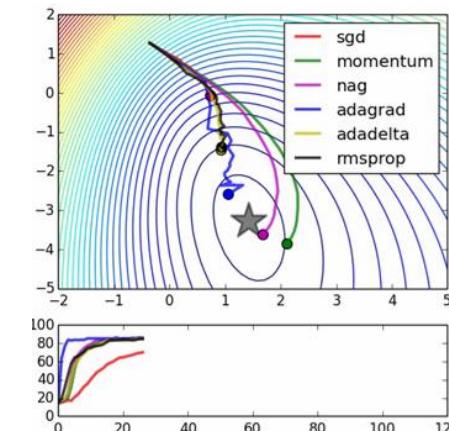
- Regularization

- Early Stopping
- L1, L2 Regularization
- Dropout



- Optimization

- SGD(Stochastic Gradient Descent)
- AdaGrad
- RMSprop
- Adam



별첨. Deep Learning의 퍼포먼스 (1/3)

■ 왜 Deep Learning의 정확도가 높은가?

- 사람이 인식하는 사물과 생각을 모두 언어로 표현할 수 있는가? No!
- 언어의 해상도 < 인식의 해상도



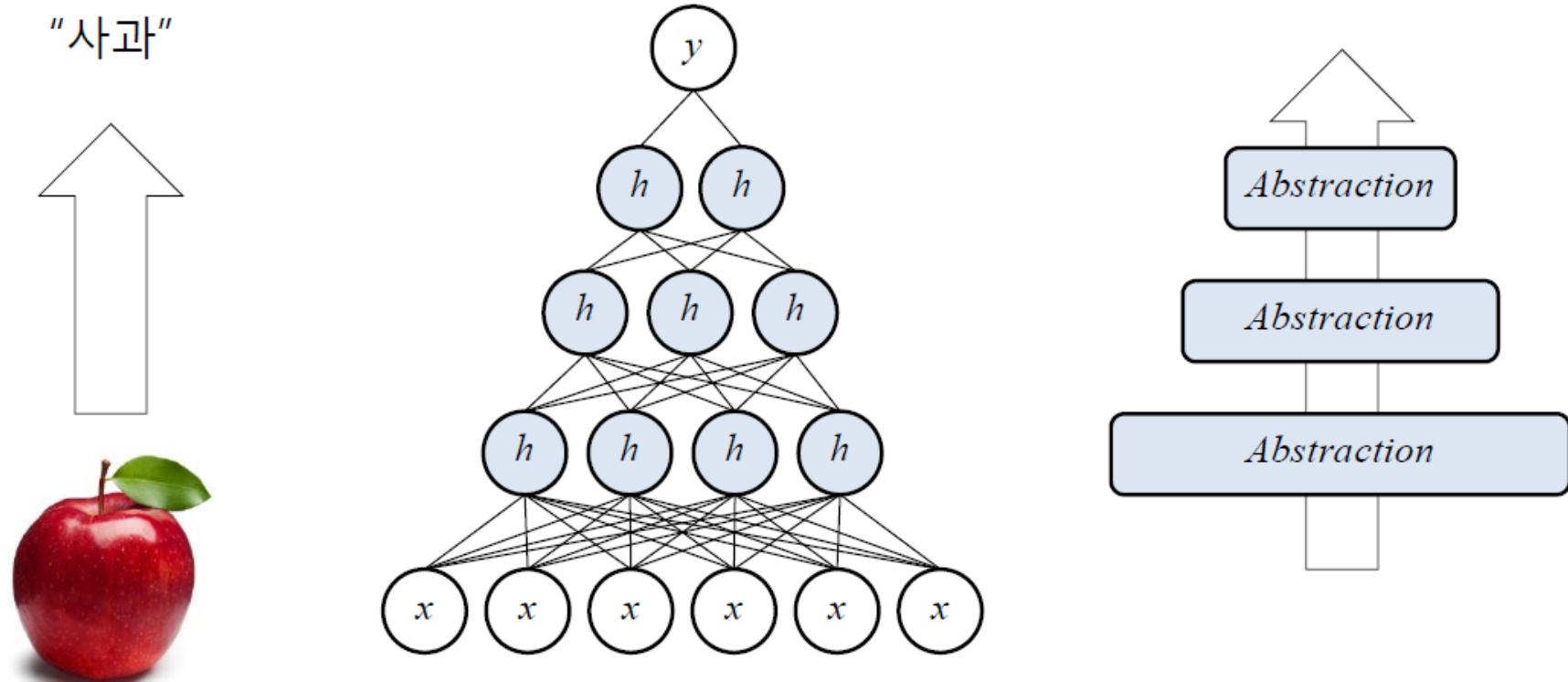
“사과”를 표현하면?

- 색상 = “빨강”
- 모양 = “둥근”
- 잎사귀 = “있다”
- 점 = “있다”
-

별첨. Deep Learning의 퍼포먼스 (2/3)

■ 왜 Deep Learning의 정확도가 높은가?

- 데이터 학습을 통하여 층(Layer)별 다양한 표현을 포착하고 추상화함 → 인식의 해상도 포착
- 은닉층이 많을 수록 다양한 표현 포착 → 패턴 인식 수준이 높아짐

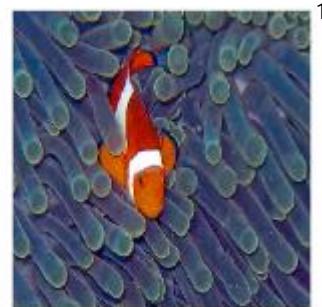




■ 왜 Deep Learning의 정확도가 높은가?

- 데이터 학습을 통하여 층(Layer)별 다양한 표현을 포착하고 추상화함 → 인식의 해상도 포착
- 은닉층이 많을 수록 다양한 표현 포착 → 패턴 인식 수준이 높아짐

Test image



1)

☞ 위 물고기 이미지 인식에 가장 큰 영향을 끼친 학습 데이터는?

- 일반 Machine Learning 알고리즘(e.g. SVM) 경우:



- Deep Learning 알고리즘(e.g. GoogleNet) 경우:

VS.

