1장 딥러닝이란?

1.1 인공지능, 머신러닝, 딥러닝

관계 1: 연구 분야

그림 출처: 교보문고(에이지 오브 머신러닝)

관계 2: 역사

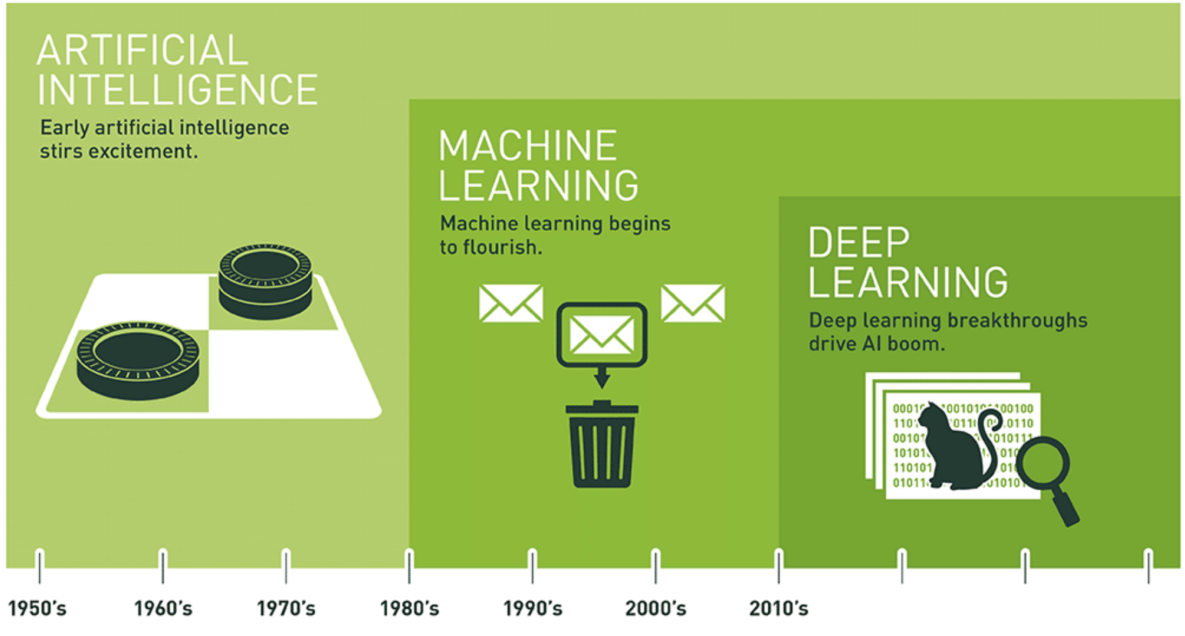


그림 출처: NVIDIA 블로그

인공지능

- (1950년대) 컴퓨터가 생각할 수 있는가? 라는 질문에서 출발
- (1956년) 존 맥카시(John McCarthy)
    - 컴퓨터로 인간의 모든 지능 활동 구현
    - 정의: 사람의 지능적 작업을 컴퓨터로 자동화하는 노력
- (1980년대까지) **학습**(러닝)이 아닌 가능한 모든 규칙을 지정하는 **심볼릭 AI**(symbolic AI) 연구
    - 모든 가능성을 논리적으로 전개하는 기법
    - 서양장기(체스) 등에서 우수한 성능 발휘
    - 반면에 이미지 분류, 음석 인식, 자연어 번역 등 보다 복잡한 문제는 제대로 다루지 못함.
- 이후 머신러닝 등장

머신러닝

- 머신러닝 시스템:
    - 명시적인 규칙(명령문)만을 수행하지 않음.
    - 데이터로부터 특정 통계적 구조 학습 후 지정된 작업을 스스로(자동으로) 완수할 수 있는 규칙 생성
    - 예제: 사진 태그 시스템. 태그 달린 사진 데이터셋을 학습한 후 자동으로 사진의 태그 작성.
- 1990년대 본격적으로 발전
    - 보다 빨라진 하드웨어와 보다 커진 데이터셋의 영향

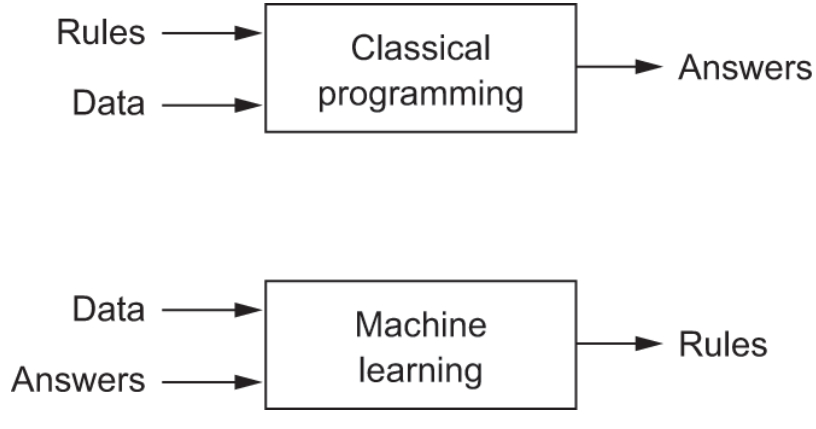


그림 출처: Deep Learning with Python(Manning MEAP)

머신러닝 대 통계학

- 머신러닝의 기초는 통계학. 1학기 내용 참조.
- 하지만 아주 큰 데이터(빅데이터)를 단순히 통계학적으로 다룰 수는 없음.
- 특히 딥러닝의 경우 수학적, 통계적 이론 보다는 공학적 접근법이 보다 중요해짐.
    - 소프트웨어와 하드웨어의 발전에 보다 의존함

데이터 표현법 학습

- 데이터 **표현**(data representation): 특정한 방식으로 구현된 데이터
- 예제: 컬러 이미지 표현법
    - 빨간색-초록색-파란색을 사용하는 RGB 방식 또는
    - 색상-채도-명도를 사용하는 HSV 방식
- 주어진 과제에 따라 적절한 표현법 선택해야 함
    - 컬러 사진에서 빨간색 픽셀만을 선택하고자 할 때: RGB 방식 활용
    - 컬리 이미지의 채도를 낮추고자 할 때: HSV 방식 활용

머신러닝 모델

- 필요 사항
    - **입력 데이터셋**: 음성 인식 모델을 위한 음성 파일, 이미지 태깅 모델을 위한 사진 등.
    - **기대 출력값**: 음성 인식 작업의 경우 사람이 직접 작성한 글, 이미지 작업의 경우 '강아지', '고양이', 등의 사람이 직접 붙힌 태그.
    - **알고리즘 성능측정법**: 출력 예상값과 기대 출력값 사이의 거리(차이) 측정법. 거리를 줄이는 방향으로 알고리즘에 사용되는 파라미터를 반복 수정하는 과정을 **학습**이라 부름.
- 역할: 입력 데이터를 적절한 표현으로 변환한 후, 변환된 테이터셋으로부터 과제 해결을 위한 적절한 수학적, 통계적 규칙 찾기

예제: 선형 분류

1. 왼쪽 그림: 입력 데이터셋
2. 가운데 그림: 부적절한 좌표 변환. 분류 과제 해결에 적합하지 않음.
3. 오른쪽 그림: 적절한 좌표 변환. 분류 과제를 보다 효율적으로 해결할 수 있음.

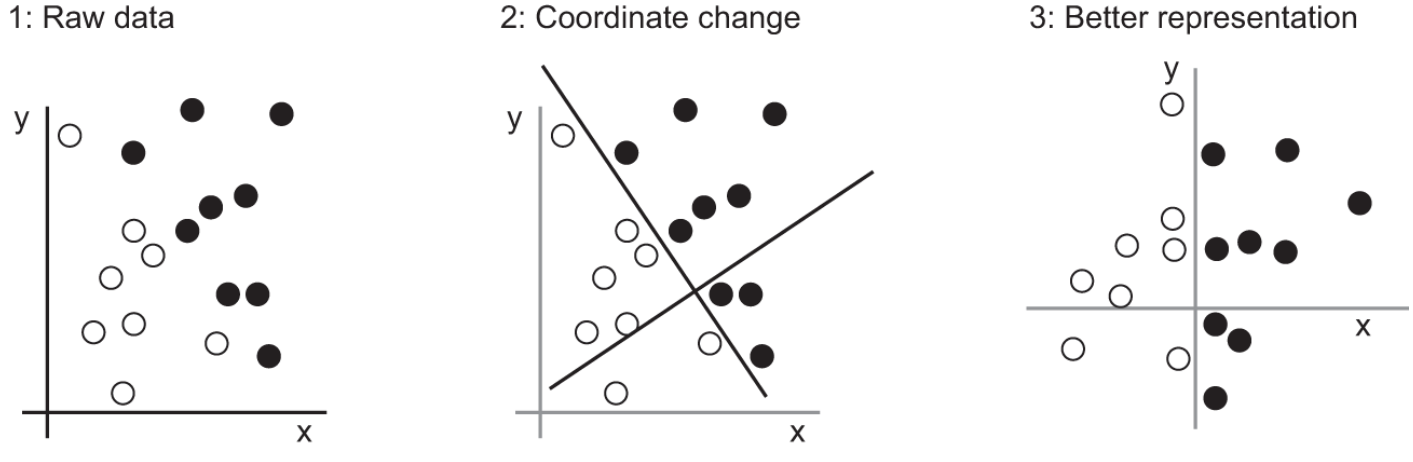


그림 출처: Deep Learning with Python(Manning MEAP)

데이터 변환 수동화의 어려움

- 위 예제의 경우 수동으로 데이터 변환 방식을 어렵지 않게 알아낼 수 있음.
- 반면에 손글씨 숫자 인식(MNIST)의 경우 간단하지 않음(아래 그림 참조).

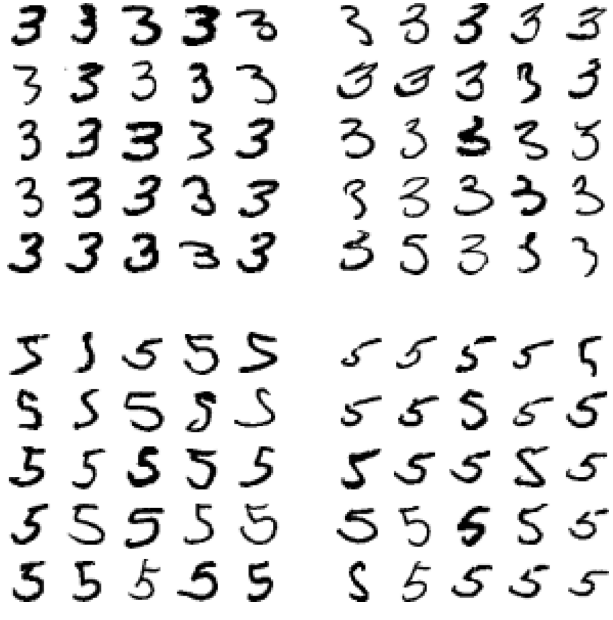


그림 출처: 핸즈온 머신러닝(2판)

데이터 변환 자동화

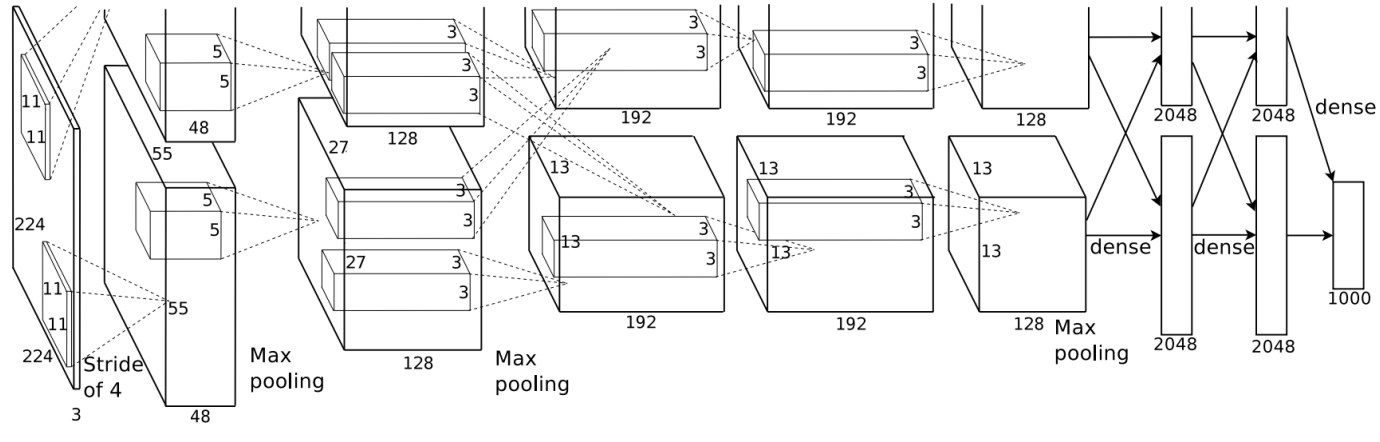
- 머신러닝 모델 학습: 보다 유용한 데이터 표현으로 변환하는 과정을 자동으로 찾는 과정
- **데이터 표현의 유용성**에 대한 기준: 주어진 과제 해결을 위한 보다 쉬운 규칙 제공
- 변환 방식 종류
    - 좌표 변환, 픽셀 개수, 닫힌 원의 개수, 선형/비선형 변환, 이동, ...
    - 기본적으로 주어진 문제에 따라 다른 변환 방식 활용

가설 공간

- 주어진 문제에 가장 적절한 변환을 머신러닝 알고리즘 스스로 알아내기는 기본적으로 불가능.
- **가설 공간**: 프로그래머에 의해 지정된 함수들의 집합
- 머신러닝 알고리즘: 가설공간 내에서 적합한 변환 함수 탐색
- 예제: 위 2차원 좌표 변환 문제의 가설 공간은 '모든 가능한 좌표 변환 함수'들의 집합

딥러닝의 '딥'(deep)이란?

- '딥'(deep)이란?: **데이터 표현의 연속적 변환**을 지원하는 여러 개의 '층'(layer)을 활용한 학습
- 즉 **계층적 표현 학습**을 지원하는 머신러닝을 딥러닝이라 부름.
- 딥러닝 모델의 깊이 = 계층으로 쌓아 올린 층의 높이
    - 수 십개 또는 수 백개의 층으로 구성된 모델 존재
    - 모든 층에서 데이터 표현의 변환이 **자동**으로 이루어지는 것이 핵심!
- 샐로우 러닝(shallow learning): 한 두 개의 층만 사용하는 학습



<그림 참조: ImageNet Classification with Deep Convolutional Neural Networks>

신경망

- 유닛(unit): 층(layer)을 구성하는 요소. 몇 개에서 몇 십개로 이루어짐.
- 신경망(neural network): 계층적 표현 학습이 이루어지는 모델
    - **계층적 '인풋-투-타깃'(input-to-target) 변환** 학습 모델
    - 예제: 숫자 이미지 $\Rightarrow \cdots \Rightarrow$ 숫자
- 단순하지만 매우 강력한 결과를 생산하는 아이디어! **뇌 과학과 아무 상관 없음!**
- 예제: 손글씨 숫자 인식

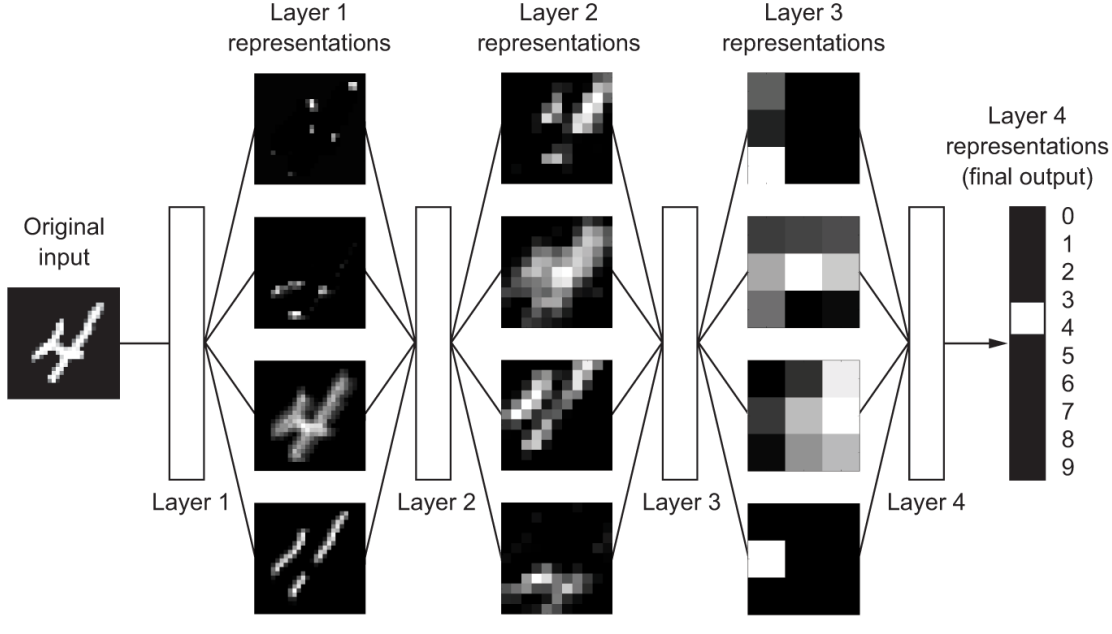


그림 출처: Deep Learning with Python(Manning MEAP)

딥러닝 작동 원리

딥러닝 모델의 작동 원리를 이해하려면 다음 세 개념에 집중해야 함

- 가중치(weight)
- 손실 함수(loss function)
- 역전파(backpropagation)

A. 가중치

- 데이터 표현의 변환에 사용되는 **파라미터**(parameter)
- 학습: 적절한 가중치를 모든 층에 대해 동시에(!) 찾는 과정
    - 하나의 가중치가 변하면 모든 다른 가중치도 변함!
- 많게는 수 천만 개의 가중치를 학습해야 함.

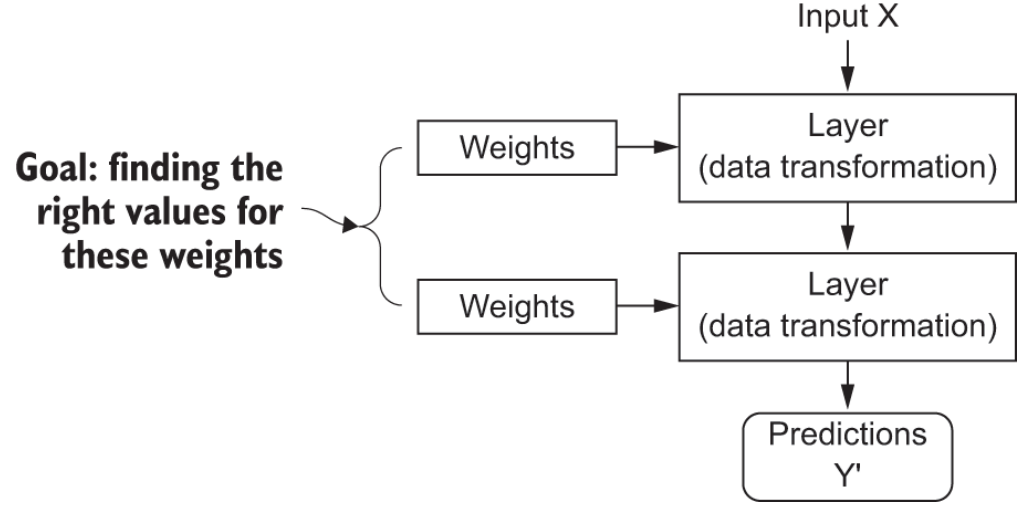


그림 출처: Deep Learning with Python(Manning MEAP)

예제: 다층 퍼셉트론(MLP, multilayer perceptron)

- 가장 간단한 형태의 신경망

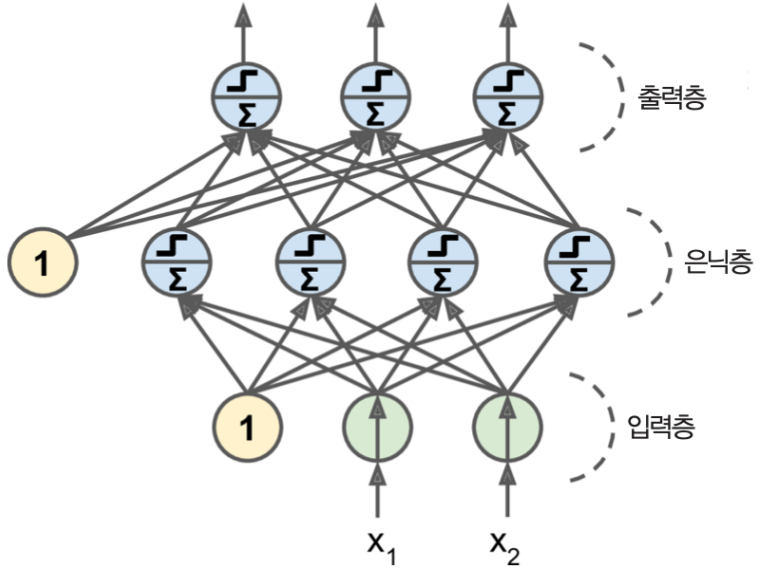


그림 출처: 핸즈온 머신러닝(2판)

유닛(unit) 작동 원리

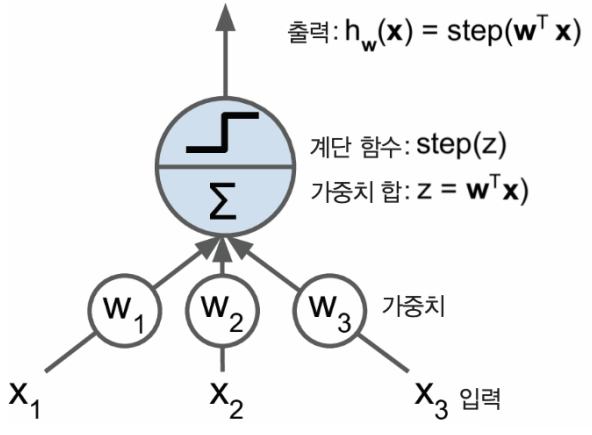


그림 출처: 핸즈온 머신러닝(2판)

B. 손실 함수

- 신경망의 출력값(output)과 타깃(target) 사이의 거리 측정. 가중치에 의존.
- **목적 함수**(objective function) 또는 **비용 함수**(cost function)라고도 불림
- 손실 함수의 반환값을 학습 과정에서 성능 평가용 피드백으로 활용.

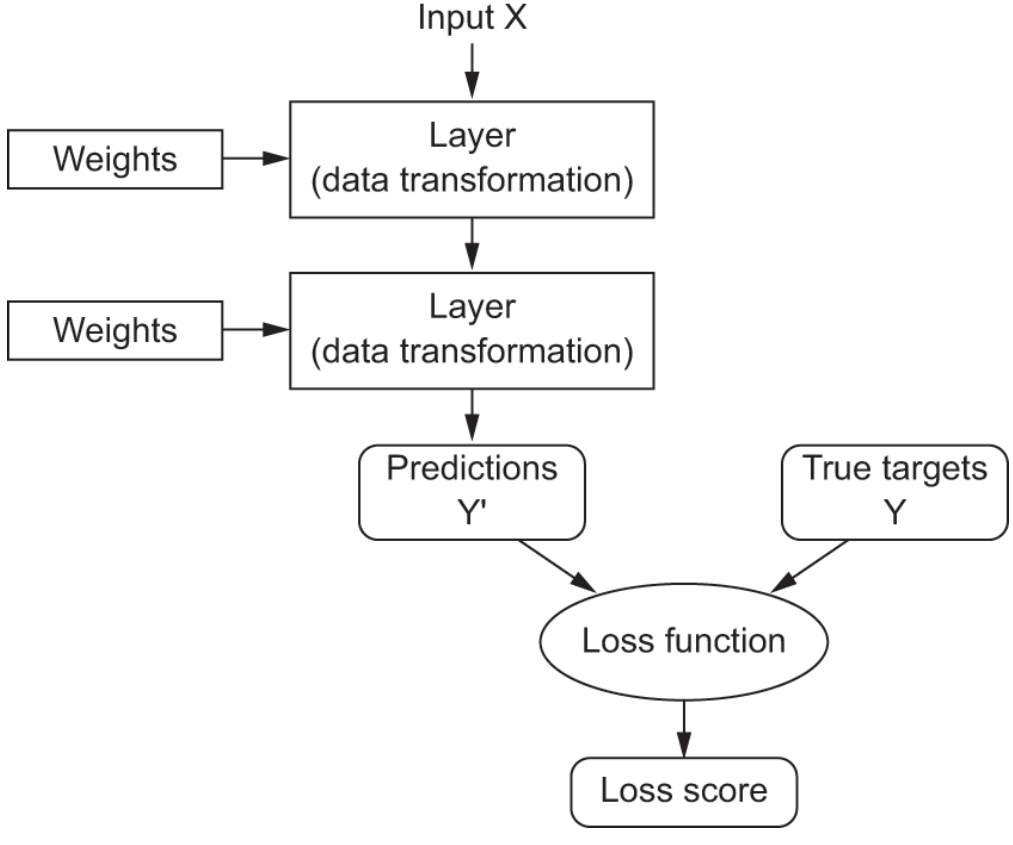


그림 출처: Deep Learning with Python(Manning MEAP)

C. 역전파

- 역전파(backpropagation) 알고리즘: 경사하강법에 기초하여 손실함수의 출력값과 타깃 사이의 거리를 좁혀주는 알고리즘
- 옵티마이저(optimizer): 역전파 알고리즘을 구현한 프로그램.
    - 모든 가중치 무작위 초기화: 결과적으로 손실값 매우 높음.
    - 훈련 반복: 손실값이 낮아지도록 가중치 조절.

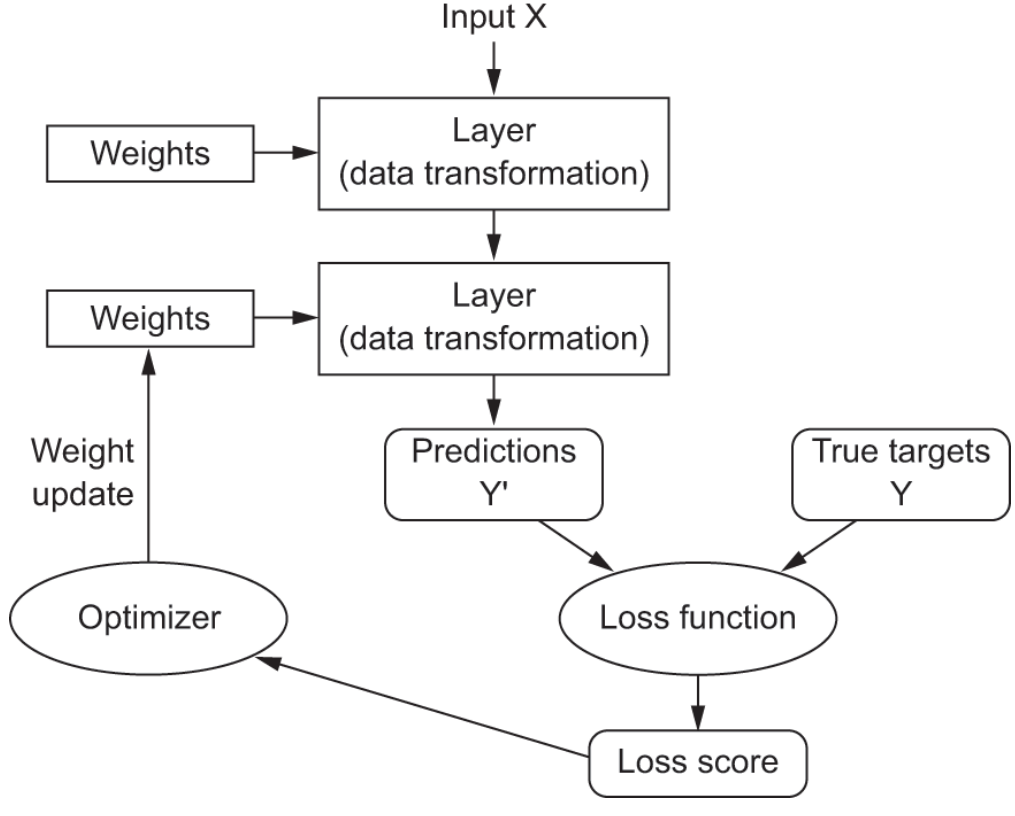


그림 출처: Deep Learning with Python(Manning MEAP)

딥러닝의 지금까지 성과

- 사람과 비슷한 수준의 이미지 분류, 음성 인식, 필기 인식, 자율 주행
- 상당한 성능의 기계 번역, TTS(text-to-speech) 변환
- 구글 어시스턴트, 아마존 알레사 등의 디지털 도우미
- 향상된 광고 타게팅, 웹 검색
- 자연어 질문 대처 능력
- 초인류 바둑 실력(2013 알파고)

전망

- 단기적으로 너무 높은 기대를 갖는 것은 위험함.
    - 실망할 경우 AI에 대한 투자가 급속도로 줄어들 수 있음.
    - 1970년대와 1990년대 1차, 2차 AI 겨울(AI winter) 경험
- 2020년대 초반 현재 중요한 문제에 본격적으로 딥러닝 적용되고 있지만 대중화는 아직.
- 1995년의 인터넷 처럼 앞으로 딥러닝의 가져올 영향에 대해 제대로 알 수 없음.

1.2 딥러닝 이전: 머신러닝의 역사

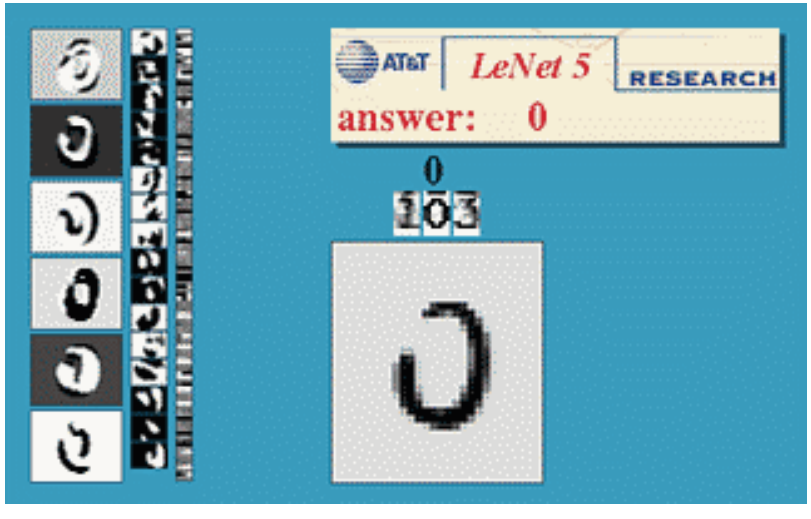
- 산업계에서 사용되는 머신러닝 알고리즘의 대부분은 딥러닝 알고리즘이 아님.
- 훈련 데이터가 너무 적거나, 딥러닝과 다른 알고리즘이 보다 좋은 성능 발휘 가능.

확률적 모델링

- 나이브 베이즈 알고리즘(Naive Bayes algorithm)을 활용하는 분석 기법이 대표적임.
- 베이즈 정리(Bayes theorem)에 기초하는 전통적인 기법
- 1950년대 부터 컴퓨터 없이 적용 시작
- 베이즈 정리 등 확률론의 기초는 18세기에 시작
- 예제: 로지스틱 회귀(logistic regression)

초창기 신경망

- 기본 아이디어: 1950년대부터 연구됨.
- LeNet 합성곱 신경망: 손글씨 숫자 이미지 자동 분류 시스템
    - 1989년 벨 연구소(Bell Labs)의 얀 르쿤(Yann LeCun)



< 그림 출처: LeNet-T CNN >

커널 기법

- 1990년대: 서포트 벡터 머신(SVM) + 커널 기법
- 초창기 신경망 성능을 뛰어 넘음.
- 한계
    - 대용량 데이터셋 처리에 부적합(매우 느림)
    - 이미지 분류 등 지각 문제 해결 어려움
- **특성 공학**(feature engineering)에 약함
    - 유용한 데이터 표현으로의 변환을 수동을 해결해야 함

결정트리, 랜덤 포레스트, 그레이디언트 부스팅 머신

- (2000년대) 결정트리: 입력값을 순서도 형식으로 특정 기준으로 분류하는 방식
- 랜덤 포레스트: 2010년 경까지 커널 기법보다 선호됨.
- 그레이디언트 부스팅 머신: 2014년 경 가장 선호되는 앙상블 학습 기법
    - 지각 문제 이외의 경우 여전히 가장 성능이 좋은 모델 중 하나임.

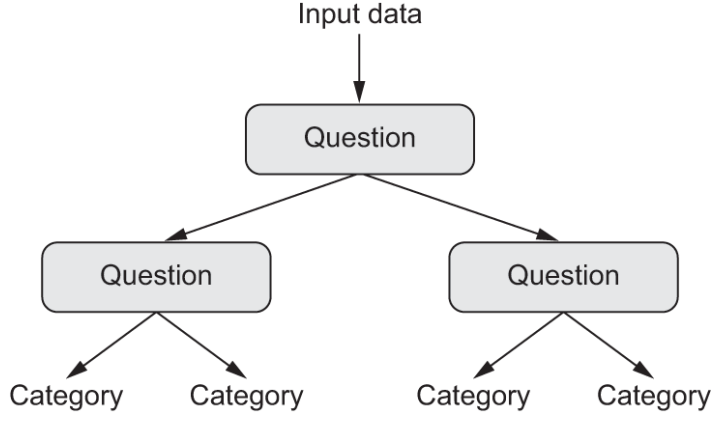


그림 출처: Deep Learning with Python(Manning MEAP)

딥러닝의 본격적 발전

- 2011년: GPU를 활용한 딥 신경망 훈련 시작
- 2012년: ImageNet Challenge(이미지 분류 경진대회)의 획기적 성공
    - 2011년 최고 성능: 74.3%의 정확도
    - 2012년 합성곱 신경망(convnet)의 최고 성능: 83.6%의 정확도
    - 2015년 최고 성능: 96.4%의 정확도
    - ImageNet Challenge 대회 더 이상 진행되지 않음.
- 2015년 이후: 많은 문제 영역에서 SVM, 결정트리 등을 딥러닝 모델로 대체함.

딥러닝의 특징

- 자동화된 데이터 표현의 변환, 즉 특성 공학 자동화
- 층을 거치면서 점진적으로 더 복잡한 데이터 표현을 만들어 냄.
- 모든 과정의 데이터 표현의 변환, 즉 모든 층에 대한 특성 공학 스스로 해결

최근 머신러닝 분야의 동향 1

- 2019년 캐글(Kaggle) 경진대회에서 상위팀이 사용한 도구 설문조사 결과

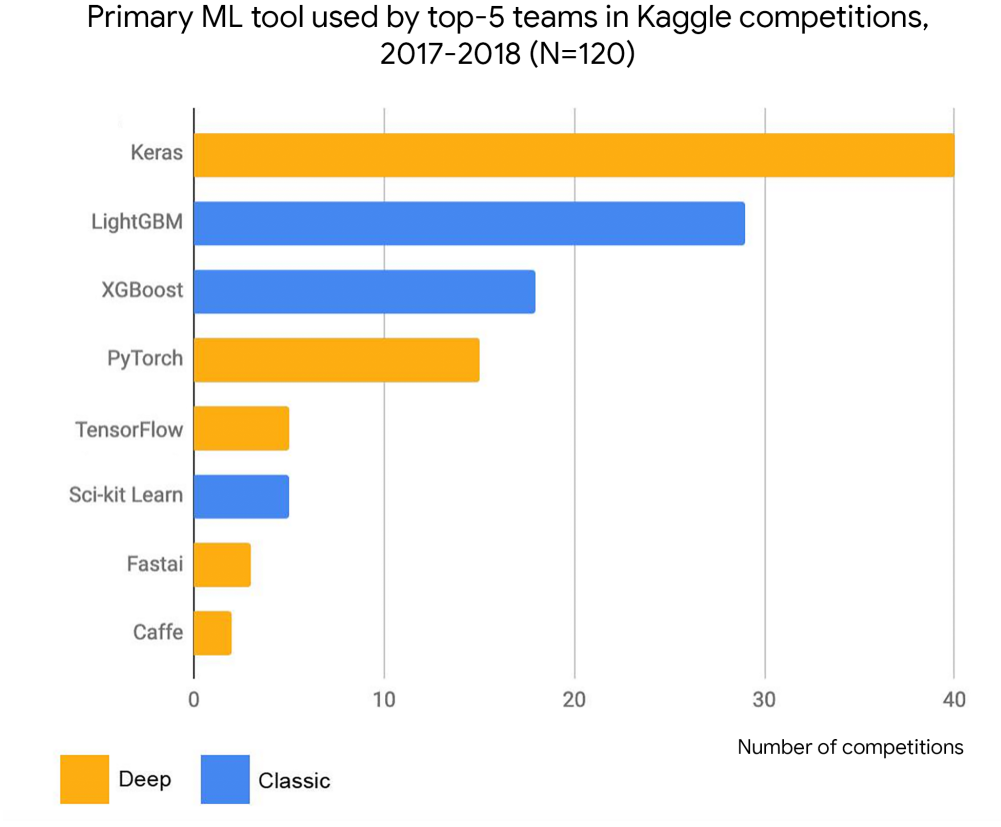


그림 출처: Deep Learning with Python(Manning MEAP)

최근 머신러닝 분야의 동향 2

- 데이터과학 일반에서 가장 많이 사용되는 도구(캐글 설문조사 2020)

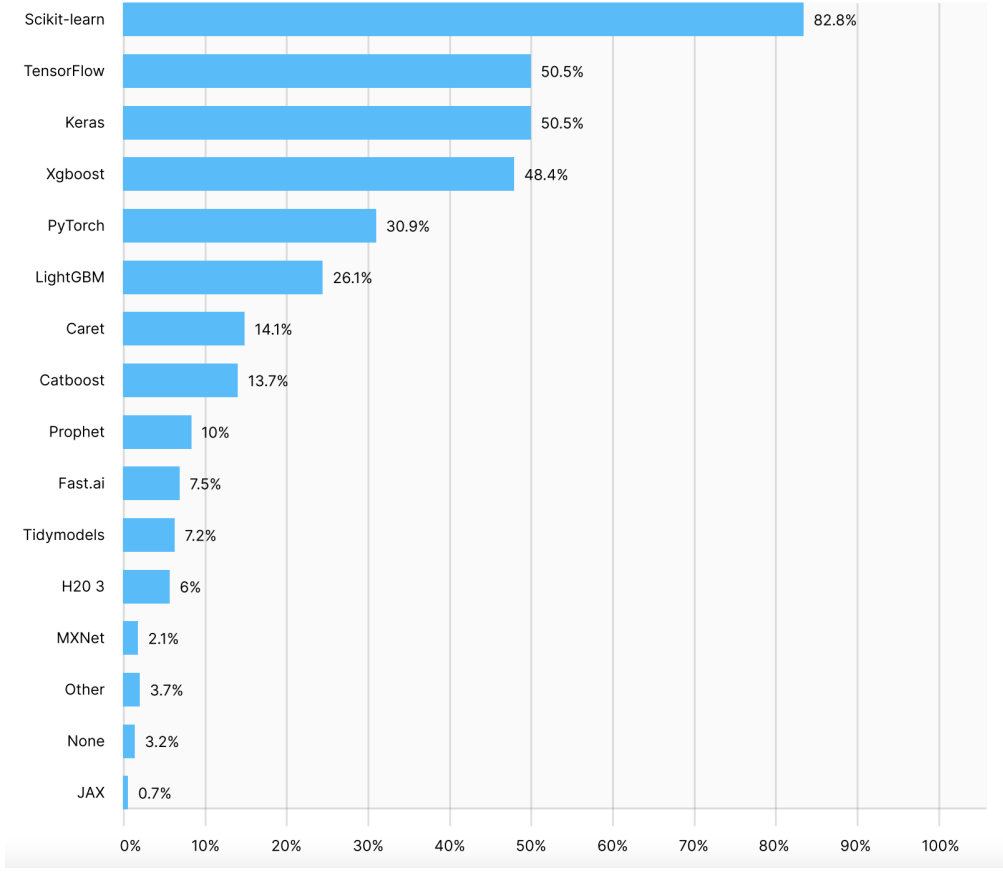


그림 출처: www.kaggle.com/kaggle-survey-2020(20쪽)

1.3 딥러닝 발전 동력

딥러닝 주요 요소

- 컴퓨터 비전, 자연어 인식 분야가 2012년 이후 획기적으로 발전하였음.
- 획기적 발전에 기여한 아래 기법은 하지만 1990년대에 제시됨
    - 1990년: 합성곱 신경망(convnet, convolutional neural network)과 역전파(backpropagation)
    - 1997년: LSTM(Long Short-Term Memory)
- 2010년대에 딥러닝의 급격한 발전에 기여한 세 가지 요소
    - 하드웨어
    - 데이터셋과 벤치마크
    - 알고리즘

하드웨어

- CPU: 1990년에 비해 5,000배 이상 빨라짐
- GPU(Graphical Processing Unit):
    - NVIDIA, AMD: 2000년대부터 게임용 그래픽 카드 개발에 천문학적으로 투자
    - 2007년 NVIDIA의 CUDA 개발: GPU를 위한 프로그래밍 인터페이스. 다량의 행렬 계산을 병렬처리 가능해짐.
    - 2011년: 신경망용 CUDA 개발.
- TPU(Tensor Processing Unit): 2016년 구글이 소개한 딥러닝 전용 칩.
    - GPU보다 훨씬 빠르고 에너지 효율적임.
    - 2020년에 3세대 TPU 카드 발표. 1990년의 최고 슈퍼컴퓨터보다 10,000배 이상 빠름.
    - 2020년 최고의 슈퍼컴퓨터 = 27,000 개의 NVIDIA GPUs = 10개의 pod 성능 (1 pod = 1024개의 TPU 카드)

데이터

- 인터넷과 저장장치의 발전으로 인한 엄청난 양의 데이터 축적
    - **무어의 법칙**(Moore's law): 2년마다 반도체 집적회로의 성능이 2배로 향상됨.
- Flickr(이미지), YouTube(동영상), Wikipedia(문서) 등이 컴퓨터 비전과 자연어 처리(NLP)의 혁신적 발전의 기본 전제조건이었음.
- 벤치마크(성능비교)의 활성화
    - ImageNet 데이터셋: 140만 개의 이미지와 손으로 작성된 1,000개의 클래스 태그
    - lamgeNet Challenge, Kaggle Competitions 등과 같은 경진대회

알고리즘

- 2000년대 후반까지 딥러닝 네트워크를 효율적으로 훈련시킬 수 있는 알고리즘 부재(역전파 문제 미해결)
- 2009-2010: 주요 알고리즘 개선
    - 보다 좋은 신경망 층에 사용되는 활성화 함수
    - 보다 좋은 가중치 초기화
    - 보다 좋은 옵티마이저(RMSProp, Adam 등)
- 2014-2016: 역전파에 도움되는 다양한 기법 개발
    - 배치 정규화(batch normalization), 잔차 연결(residual connection), 깊이별 분리 합성곱(depthwise separable convolution) 등
- 현재: 수십 개의 층을 가지며 수천 만개의 가중치(파라미터)를 갖는 깊은 층으로 구성된 신경망 네트워크 훈련 가능

투자

- 2013년 이후 투자가 획기적으로 증가함

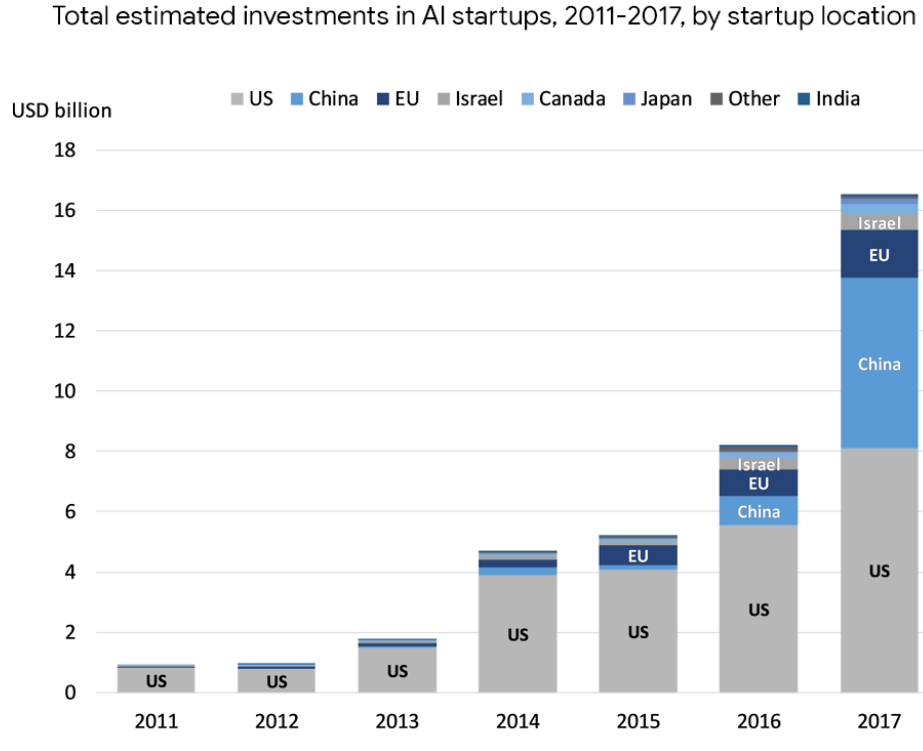


그림 출처: OECD estimate of total investments in AI startups

딥러닝의 대중화

- 이전: C++, CUDA 등을 이용한 어려운 프로그램을 구현할 수 있었어야 함.
- 지금: 파이썬 기초 프로그래밍 수준에서 시작 가능
    - Sci-kit Learn, Theano, Tensorflow, Keras 등의 라이브러리 활용

딥러닝의 미래

- 여전히 딥러닝 혁신적 발전이 진행중임.
    - 최근의 가장 큰 혁신: 트랜스포머(transformer) 기법를 이용한 자연어 처리
- 20년 뒤엔 알 수 없음. 하지만 딥러닝 발전의 토대를 이룬 아래 요소는 계속해서 활용될 것으로 기대함.
    - 단순함: 특성 공학의 자동화로 인해 모델 생성과 훈련이 단순화됨.
    - 확장성: GPU 또는 TPU 등을 이용한 병렬화가 가능하기에 무어의 법칙을 최대한 활용할 수 있음. 작은 크기의 데이터 배치(묶음)로 나눈 후 훈련 반복을 병렬화하면 임의의 크기의 데이터셋을 이용한 훈련이 가능해짐.
    - 다용도와 재사용성: 추가 입력된 데이터로 훈련을 이어갈 수 있음. 또한 잘 훈련된 모델을 다른 용도의 모델 훈련에 재활용할 수 있음. 이는 또한 아주 작은 데이터셋을 대상으로 딥러닝 모델을 적용할 수 있도록 해줌.