



1 주 차
키 워 드
보 고 서



14기 A조 양진미



CONTENTS

딥러닝	3
디지털 리터러시	3
머신러닝	4
디지털 트랜스포메이션	4
추상화	4
GPT-3	5
유닉스	6
컴파일러	6
NLP	7
모뎀	7
C언어	7
3A	8
리눅스	9
컴퓨터비전	9
강화학습	9
자율주행차	10
컴퓨터셔널 핑킹	10
자동화	11
앨런 튜링	11
오브젝트 디텍션	11
문제해결능력	12
파이썬	13
AI	13
안드로이드	14
깃 허브	14
자바	14



Deep Learning

인공신경망을 잇는 기계학습법. 딥러닝은 사물이나 데이터를 군집화하거나 분류하는 데 사용하는 기술이다.

컴퓨터가 사진만으로 개와 고양이를 구분하지 못함에, 많은 데이터를 컴퓨터에 입력하고 비슷한 것끼리 분류하도록 하는 '기계학습(Machine Learning)'이 고안되었고, 이를 '어떻게 분류할 것인가'를 놓고 생긴 많은 알고리즘 '의사결정나무', '베이지안망', '서포트벡터머신(SVM)', '인공신경망' 등의 방법들 중, 인공신경망의 한계를 극복하기 위해 제안된 Machine Learning 방법이다. 처음 제안된 때는 1980년 캘리포니아 대학 심리학자와 컴퓨터 관련 학자들의 신경망 연구를 요약한 <PDP>에서부터 이고, 제프리 힌튼 교수가 RBM(2004년)이라는 새로운 딥러닝 기반의 학습 알고리즘을 제안하면서부터 주목받기 시작했다. 이후, 드롭아웃이라는 알고리즘도 등장하여 고질적으로 비판받는 과적합 등도 해결 가능해졌다.

딥러닝의 핵심은 분류를 통한 예측이다. 수많은 데이터 속에서 패턴을 발견해 컴퓨터가 데이터를 나눈다. 이의 분별 방식은 '지도 학습(supervised learning)'과 '비지도 학습(unsupervised learning)'으로 나뉘는데, 지도 학습은 컴퓨터에 먼저 정보를 가르치는 방법이고, 비지도 학습은 이 배움의 과정이 없이 컴퓨터가 스스로 학습하게 하는 것이다. 기존 기계학습 알고리즘은 대부분 지도 학습에 기초하며, 비지도 학습은 지도 학습과 비교할 때, 진보한 기술이고, 이미 구글은 현재 비지도 학습 방식으로 고양이 동영상상을 식별하는 딥러닝 기술을 개발한 상태이다.

Digital Literacy

디지털 문해력을 의미한다. 디지털로 기록되고 저장된 정보를 사용하여 만들어지거나 전송된 여러 양상을 통해 세상의 의미를 인코딩하고 디코딩하는 문화적 관습을 의미한다. 네트워크 사회에서는 참여자들의 의미와

아이덴티티(identity)를 협상하는 스킬, 태도, 기질을 아우르며 기술을 통한 개인의 문법, 작문, 글쓰기 등을 포함한다.

즉, 디지털 리터러시란 글을 읽고 쓸 줄 아는 것처럼 디지털을 활용할 줄 아는 역량을 말한다. 이 용어는 1997년 Paul

Gilster의 <Digital Literacy>에서 처음 소개 되었고, "컴퓨터를 통해 다양한 출처로부터 찾아낸 여러 가지 형태의 정보를 이해하고 자신의 목적에 맞는 새로운 정보로 조합해 냄으로써 올바르게 사용하는 능력(Ability to understand and use information multiple formats from a wide range of sources when it is presented via

computers)"라고 정의했다. 90년대 말 2000년대 초에는 협의의 의미로 디지털 리터러시를 정의했으나,

2010년대에서부터 단순한 기능과 능력을 넘어서 태도와 마인드까지 포함하는 역량의 관점으로 변화되기 시작하였다.

브리티시 컬럼비아는 디지털 리터러시를 '연구 및 정보 리터러시(Research and Information Literacy)', '비판적 사고력, 문제해결, 의사 결정 (Critical Thinking, Problem Solving, and Decision Making)', '창의성 및 혁신성 (Creativity and Innovation)', '디지털 시민의식 (Digital Citizenship)', '의사소통 및 협동 (Communication and Collaboration)', '기술 활용 및 개념 (Technology Operations and Concepts)' 의 6가지 영역으로 나누고 있다.



Machine Learning

인공지능의 한 분야로, 경험적 데이터를 기반으로 학습을 하고 예측을 수행하고 스스로의 성능을 향상시키는 시스템과 이를 위한 알고리즘을 연구하고 구축하는 기술이다. 머신러닝은 패턴인식과 컴퓨터 학습 이론의 연구로부터 진화하였고, 1959년 아서 사무엘이 '컴퓨터에 명시적인 프로그램 없이 배울 수 있는 능력을 부여하는 연구 분야'로 정의하였다. 즉, 사람이 학습하듯이 컴퓨터에도 데이터들을 제공하여 학습하게 함으로써 새로운 지식을 얻어내는, '기계가 스스로 학습하여 패턴을 찾는 방식'을 의미한다.

머신 러닝은 명시적인 알고리즘을 설계하기 어렵거나 프로그래밍하기 어려운 작업을 해결하기 위해 주로 사용된다.

대부분의 머신 러닝 알고리즘은 먼저 데이터가 생성한 잠재적인 메커니즘의 특징을 파악하여 복잡한 관계를 정량화한 후, 이 식별된 패턴을 사용하여 새로운 데이터에 대한 예측을 실시하는데, 이는 데이터로부터 유용한 규칙, 지식 표현 혹은 판단 기준 등을 추출한다는 점에서 데이터마이닝이나 통계 및 수학적 최적화 문제와 관련이 깊다.

또한, 머신러닝의 알고리즘은 학습 시스템에 정보 및 데이터를 입력하는 방법에 따라서 앞서 언급한 '지도

학습(supervised learning)'과 '비지도 학습(unsupervised learning)', 그리고 '강화 학습(reinforcement learning)' 세 가지로 나뉜다. 강화 학습의 자세한 설명은 밑에서 다시 다룰 예정이며, 머신 러닝의 기법 및 모델들로는 트리 구조 형태를 예측 모델로 사용하는 의사 결정 나무(Decision Tree), 생물의 신경 네트워크 구조와 기능을 모방하는 인공 신경망(Neural Network), 생물의 진화 알고리즘에 기반한 유전자 프로그래밍(Genetic Programming), 관측된 예를 군집이라는 부분집합으로 분배하는 군집화(Clustering), 무작위로 추출된 난수를 통해 함수값을 확률로 계산하는 몬테카를로 방법(Monte Carlo method) 등이 있다.

Digital Transformation (DT 또는 DX)

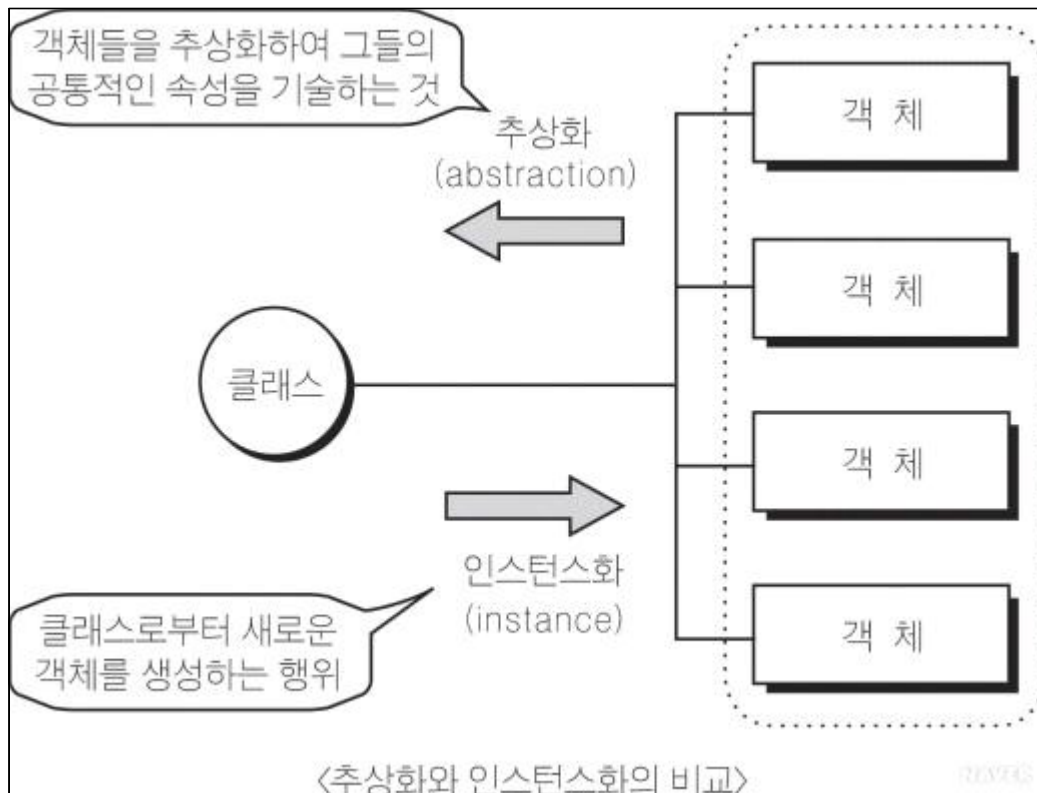
디지털 트랜스포메이션은 클라우드 컴퓨팅을 활용하여 문제를 해결하고 소유 중인 하드웨어 사용자에게 대한 의존성을 줄이고 구독 기반 클라우드 서비스에 의존성을 증대시키기 위해 새롭고 빠르고 자주 변화하는 디지털 기술을 사용하는 방식을 말한다. 즉, 디지털 기술을 사회 전반에 적용하여 전통적인 사회 구조를 혁신시키는 것이고, 일반적으로 기업에서 사물 인터넷(IoT), 클라우드 컴퓨팅, 인공지능(AI), 빅데이터 솔루션 등 정보통신기술(ICT)을 플랫폼으로 구축·활용하여 기존 전통적인 운영 방식과 서비스 등을 혁신하는 것을 의미한다. IBM 기업가치연구소의 보고서(2011)는 '기업이 디지털과 물리적인 요소들을 통합하여 비즈니스 모델을 변화시키고, 산업에 새로운 방향을 정립하는 전략'이라고 정의하고 있으며, 디지털 전환(digital transformation)을 위해서는 아날로그 형태를 디지털 형태로 변환하는 '전산화(digitization)' 단계와 산업에 정보통신기술을 활용하는 '디지털화(digitalization)' 단계를 거쳐야 한다.

추상화(Abstraction)

추상화(Abstraction)란, 컴퓨터 과학 분야에서 주어진 문제나 시스템의 복잡도를 단순화하여 인식하기 쉽게 만드는 개념화 작업을 의미한다. 핵심 요소를 잘 파악하여 필요 이상으로 복잡한 요소들을 결합하거나 단순화하고, 속성의



일부분만으로 주어진 대상을 간결하고 명확하게 표현하는 것을 의미하며, 이는 방식에 따라 데이터 추상화(data abstraction)와 절차 추상화(procedural abstraction)로 나뉘는데, 데이터 추상화는 하나의 데이터와 관련된 조작 및 표현 유형을 결합하는 방식을 의미한다. 이 기법은 복잡한 문제나 시스템을 이해하는 데 굉장히 중요한 요소이다.



GPT-3

GPT-3(Generative Pre-trained Transformer 3)는 딥러닝을 사용하여 인간과 유사한 텍스트를 생성하는 자동 회귀 언어 모델을 의미한다. 이는 샌프란시스코에 기반을 둔 인공지능 연구소 Open AI의 GPT 시리즈의 3세대 언어 예측 모델이며, GPT-3의 정식 버전은 1750억 개의 머신 러닝 매개 변수 용량을 가지고 있는 즉, Open AI사가 개발한 인공 일반 지능 모델이라고 생각하면 된다. GPT-3는 딥러닝을 이용해 인간다운 텍스트를 만들어내는 자기회귀 언어 모델이며, 각종 언어 관련 문제풀이, 랜덤 글짓기, 간단한 사칙연산, 번역, 주어진 문장에 대한 간단한 웹 코딩이 가능하다. 또한 GPT-3가 생성하는 본문의 질이 매우 높아, 호주 철학자 데이비드 찰머스는 GPT-3를 "지금까지 생산한 AI 시스템 중 가장 흥미롭고 중요한 시스템 중 하나"라고 설명했지만, 동시에 인간이 작성한 논문과 구별하기 어려운 정도라, 31개 오픈AI 연구진과 엔지니어들은 GPT-3를 소개하는 2020년 5월 28일 논문 원본을 발표하면서 GPT-3의 잠재적 위험을 경고하고 위험 완화를 위한 연구를 요구하기도 했다.

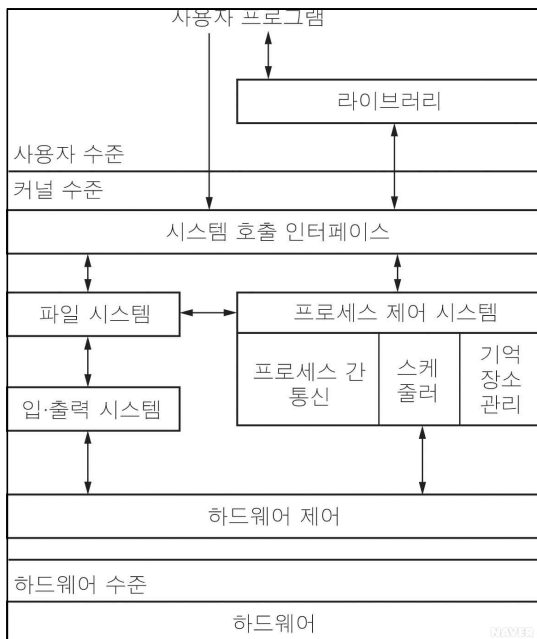
GPT-3는 방대한 양의 텍스트를 통해 다음 단어를 예측하는 방식으로 학습되었으며, 이에 따른 단점으로 새로운 정보를 수용하기 어렵고, 현실 세계의 물리적 상식은 제대로 알지 못하며, 모든 분야에서 뛰어난 것이 아니므로 효율성 또한



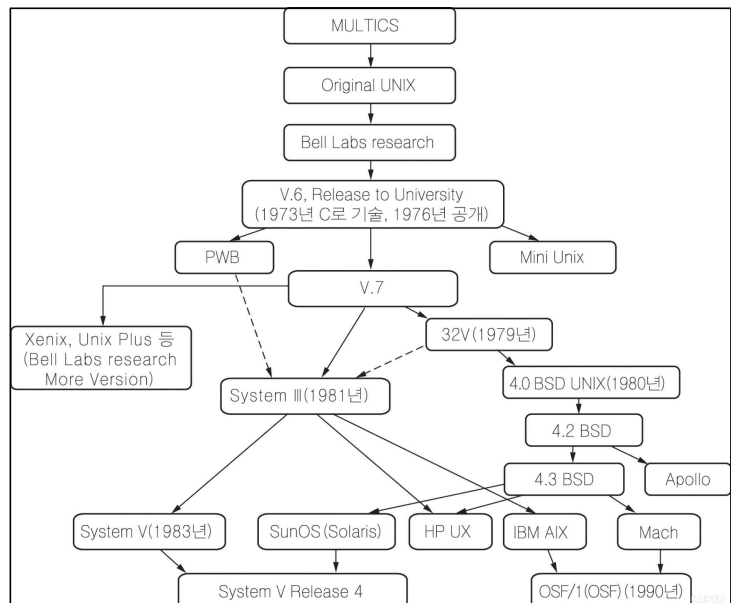
떨어진다는 점이 있다.

유닉스(Unix)

유닉스는 1969년 미국의 전신전화회사 AT&T사의 벨연구소에서 어셈블리어로 개발한 운영체제이다. 이는 이후 데니스 리치(Dennis Ritchie)가 처음으로 C언어로 다시 대부분 작성 했으며, 이는 처음으로 어셈블리가 아니라 고급 프로그래밍 언어(C언어)로 커널까지 작성된 운영체제이다. 또한 이 때문에 하드웨어로 이식하기가 쉬워졌고, 멀티 캐스팅 기술을 도입하여 여러 사용자가 동시에 다중 사용자 & 다중 태스크 실행을 가능하게 하였다. 이는 대화형의 운영 체제로, 텍스트 조작 툴, 문서 처리, 전자 메일, 프로그램 개발 외에 취급이 쉬운 파일 시스템을 갖추고 있다.



<유닉스 시스템의 커널 구조>



<유닉스의 계보>

컴파일러(Compiler)

고급언어로 쓰인 프로그램이 컴퓨터에서 수행되기 위해서, 컴퓨터가 직접 이해할 수 있는 언어로 바꾸어 주는 것을 컴파일러라고 한다. 즉, 고급언어로 쓰인 프로그램을 의미적으로 동등하게 컴퓨터에서 실행 될 수 있도록 프로그램으로 바꾸어주는 번역 프로그램이다. 컴파일을 하기 위하여 입력되는 프로그램을 원시 프로그램이라 하고 이 프로그램을 기술한 언어를 원시언어(source language)라 한다. 또한 번역되어 출력되는 프로그램을 목적 프로그램이라 하고 이 프로그램을 기술한 언어를 목적언어(object language 또는 target language)라 하고, 크로스



컴파일러(cross-compiler)는 원시 프로그램을 컴파일러가 수행되고 있는 컴퓨터의 기계어로 번역하는 것이 아니라, 다른 기종에 맞는 기계어로 번역하는 컴파일러를 말한다.

컴파일러는 컴파일하는 과정이 복잡하고 그 시간이 많이 걸리지만 한 번 컴파일하면 그대로 사용이 가능하며 반복하는 경우에 기계코드가 반복을 하기 때문에 처리시간이 매우 빠르고, 전체 실행시간 면에서 매우 효율적이며, 개발 시스템이나 교육용 시스템에서는 인터프리터가 좀 더 능률적이다.

NLP(Natural Language Processing)

NLP(Natural Language Processing)는 '자연어 처리'로, 컴퓨터를 이용해 사람의 자연어(natural language)를 분석하고 처리하는 기술이다. 자연어 처리(NLP)에는 자연어 분석, 자연어 이해, 자연어 생성 등의 기술이 사용되며, 자연어 분석은 그 정도에 따라 형태소 분석(morphological analysis) 과 통사 분석(syntactic analysis), 의미 분석(semantic analysis) 및 화용 분석(pragmatic analysis) 총 4가지로 나눌 수 있고, 자연어 이해는 컴퓨터가 자연어로 주어진 입력에 따라 동작하게 하는 기술이며, 자연어 생성은 동영상이나 표의 내용 등을 사람이 이해할 수 있는 자연어로 변환하는 기술이다. 이렇게 이루어진 자연어 처리는 인공지능의 주요 분야 중 하나로, 1950년대부터 기계 번역과 같은 자연어 처리 기술을 연구하기 시작하면서, 1990년대 이후에는 대량의 말뭉치(corpus) 데이터를 활용하는 기계 학습 기반 및 통계적 자연어 처리 기법을 주류로 하고, 최근에는 심층 기계 학습(deep learning) 기술을 기계 번역 및 자연어 생성 등에 적용하고 있다.

모뎀(Modem)

데이터 통신을 위해 컴퓨터나 단말 등의 통신 기기를 통신 회선에 접속시키는 장치이며, 모뎀(MODEM)은 약어이고, Modulator-Demodulator(변복조기) 라고 한다. 컴퓨터나 단말 등에서 나가는 디지털 신호를 아날로그 신호로 변환하는 변조기(modulator, 모듈레이터)와 들어오는 아날로그 신호를 디지털 신호로 변환하는 복조기(demodulator, 디모듈레이터)의 복합하여 역할을 수행하고, 통칭은 변조기와 복조기의 합성어이다. 이는 데이터 통신용 기기 내부의 디지털 신호와 통신 회선을 흐르는 아날로그 신호의 형태가 전혀 다르므로 이 두 신호를 상호 변환하기 위해서 변복조기(모뎀)이 필요하다.

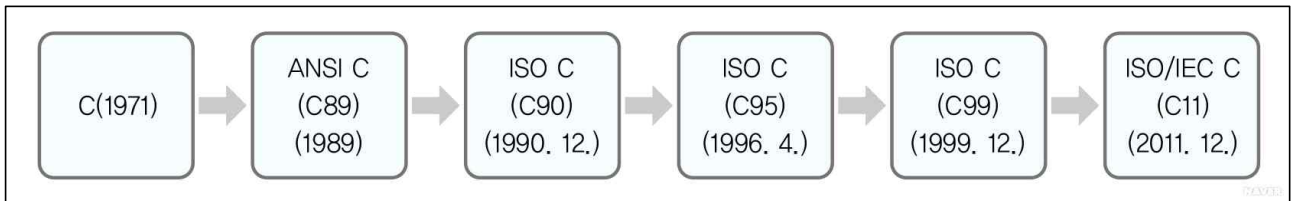
모뎀(Modem)

C언어는 벨 연구소에서 1971년경부터 리치(D.M.Ritchie) 등에 의해서 설계 개발된 시스템 기술용의 프로그래밍 언어이다. 컴퓨터의 구조에 밀착한 기초 기술이 가능한 것과 간결한 표기가 될 수 있는 것 등이 특징이며, 또한 프로그램을 기계어 명령에 가까운 유형으로 직접 기술할 수 있고, 언어를 간단하게 하여 풍부한 표준 자료집을 갖게 할

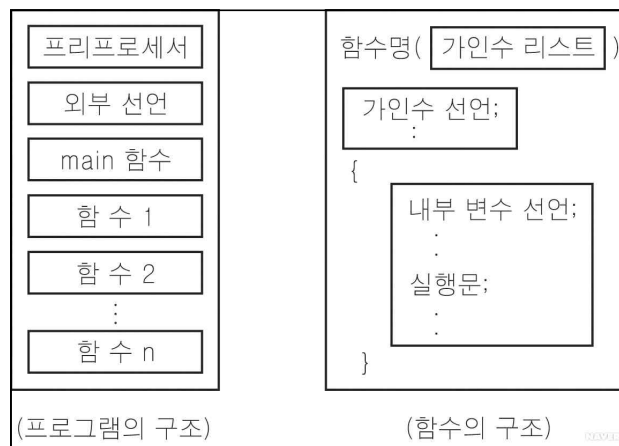


수 있으며, 연산자가 많고, 다른 기종에 프로그램 이식이 쉽다는 점도 특징이다.

C 언어는 ASCII코드 체계로 영문 소문자 집합을 바탕으로 하고 있으며, 함수(function)의 정의문들의 집합으로 구성되어 있고, 분할 컴파일을 할 수 있어 함수 정의부 밖의 외부 변수를 정의함으로써 컴파일 단위가 다른 함수의 외부 변수를 참조할 수 있게 되어 있다. 또한 프로그램의 함수 기능에는 FORTRAN의 SUBROUTINE 기능과 FUNCTION 기능을 가지고 있으며, 함수 호출시 매개변수의 값만 넘겨주는 호출 방식을 따르고 있고, 자료의 주소를 자유롭게 조작할 수 있다.



<C언어의 역사>



<C언어 프로그램의 구조>

3A

3A는 Abstraction, Algorithm, Automation 으로, Abstraction은 앞서 언급한 추상화이고, Algorithm은 알고리즘으로, 어떤 문제를 해결하기 위해 명확히 정의된 유한 개의 절차, 방법, 명령어들의 집합을 의미한다. 또한 Automation은 자동화로, 일련의 과정을 자동화 시키는 것을 의미하며, 이는 자동적으로 스스로의 과정을 진행 시키게 하는 것을 의미한다. 이 세 가지의 단어는 컴퓨터서널 댕킹(Computational Thinking)의 3가지 주요 특징으로, 밑에서 이 사고력에 대해서는 다시 언급할 예정이다.

즉, 알고리즘은 주어진 문제를 논리적으로 해결하기 위해 필요한 절차, 방법, 명령어들을 모아놓은 것이며, 추상화를 통해 복잡한 알고리즘을 간결화하고, 이를 자동화 시켜 같은 상황에 놓여질 때, 다시 시작하는 것이 아니라 자동화 되어 반복 시킬 수 있도록 하는 컴퓨팅 사고력의 기본적인 특징이다.



리눅스 (Linux)

리눅스는 1991년 리누스 토르발즈(Linus Torvalds)가 공개한 유닉스를 기반으로 한 개인컴퓨터용 공개 운영체제이다. 이는 대형 기종에서만 작동하던 운영 체제인 유닉스(UNIX)를 386 기종의 개인용 컴퓨터(PC)에서도 작동할 수 있게 만든 운영 체제이며, 인터넷을 통해 프로그램 소스 코드도 무료로 공개하여 사용자는 원하는 대로 특정 기능을 추가할 수 있게 하고, 어느 플랫폼에도 포팅이 가능한 운영체제이다. 파일구성이나 시스템기능의 일부는 유닉스를 기반으로 하면서, 핵심 커널 부분은 유닉스와 다르게 작성되어 있어, 인터넷 프로토콜인 TCP/IP를 강력하게 지원하는 등 네트워킹에 특히 강점을 지니고 있다. 또한 유닉스와 거의 유사한 환경을 제공하면서 무료라는 장점 때문에 프로그램 개발자 및 학교 등을 중심으로 급속히 사용이 확대되었다.

Computer Vision

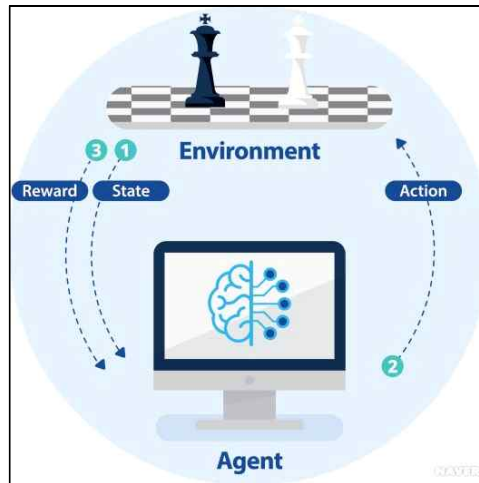
Computer Vision이란, 인공지능의 한 분야로 컴퓨터를 사용하여 인간의 시각적인 인식 능력 일반을 재현하는 연구 분야이다. 컴퓨터 시각이라고 불리우며, CV 약어로도 혼용된다. 이는 사람이나 동물 시각 체계의 기능을 컴퓨터로 구현하는 것으로, 주로 사진이나 동영상 같은 2차원 이미지에서 정보를 추출하는 컴퓨터 기술을 연구한다. 사람의 시각 체계는 망막에 맺힌 2차원 이미지에서 3차원 모델이나 구조를 추론하고, 물체를 탐지하거나 이미지 내부의 영역을 구분하는 등 눈으로 보는 것에서 다양한 정보를 추출하는 역할을 하는데, 컴퓨터 시각은 이러한 기능을 컴퓨터로 구현하는 것을 가리키며 주로 디지털카메라나 캠코더 등의 이미지 센서에 맺힌 2차원 이미지를 처리한다. 로봇이나 자율 주행 자동차와 같은 지능형 에이전트(intelligent agent) 구현에 꼭 필요하며, 위성사진 분석과 같은 작업에 활용하기도 한다. 2000년대에 들어서면서, 인터넷의 발달로 대규모 데이터의 활용이 가능해지면서 기계학습 기반 컴퓨터 시각 기술의 성능이 비약적으로 발전하였으며 자동차 번호판 인식 등의 기술까지 실용화되었다. 2010년대에 확산된 심층기계학습(deep learning) 기반의 컴퓨터 시각 기술은 잡음(noise)이 없는 사진에서는 수천 종류의 사물을 인식하는 등 한정된 상황에서 물체를 인식할 때는 사람과 유사하거나 더 뛰어난 정확도를 보였다. 현재 컴퓨터 시각 기술은 자동차 번호판 인식, 디지털카메라의 얼굴 인식, 공장에서의 제품 검사 등에 활용된다.

강화학습(Reinforcement Learning)

Reinforcement Learning이란, 기계 학습 중 컴퓨터가 주어진 상태(state)에 대해 최적의 행동(action)을 선택하는 학습 방법이다. 강화형 기계 학습은 지도형/비지도형 기계 학습에 이용되는 훈련 데이터와 달리 주어진 상태에 맞춘 행동의 결과에 대한 보상(reward)을 주는 형식으로 성능을 향상 시키는데, 주로 게임이나 로봇 제어 등에 적용된다. 강화 학습의 대상이 되는 컴퓨터 프로그램을 에이전트(agent)라고도 한다. 에이전트는 주어진 상태(state)에서 자신이 취할 행동(action)을 표현하는 정책(policy)을 수립한다. 에이전트가 최대의 보상을 받을 수 있는 정책을 수립하도록 학습시키는 것이 강화 학습의 목표이다. 강화 학습의 주요 응용 분야로는 게임과 로봇 제어이며, 1992년 IBM의 제럴드 테사우로(Gerald Tesauro)가 강화 학습을 이용하여 개발한 백개먼(Backgammon) 게임(TD-Game)은 인간 챔피언과



유사한 수준에 도달했고, 2016년 딥마인드(DeepMind)가 개발한 강화 학습 기반의 소프트웨어 알파고(AlphaGo) 또한 강화학습을 이용한 대표적인 예 이다.



<강화학습을 이용한 예시>

자율주행차

자율주행차(self-driving car, autonomous vehicle, AV, driverless car, robo-car)는 무인주행차로도 불리며, 이는 엄밀한 의미에서 사람이 타지 않은 상태에서 움직이는 무인자동차(driverless cars)와는 다르지만 실제로 혼용되어 인간의 운전없이 자동으로 주행할 수 있는 자동차를 의미한다. 자율주행차는 레이더, LIDAR(light detection and ranging), GPS, 카메라로 주위의 환경을 인식하여 목적지를 지정하는 것만으로 자율적으로 주행하는 차로, 자율주행차를 위해서는 고속도로 주행 지원 시스템(HDA)을 비롯해 후측방 경보 시스템(BSD), 자동 긴급 제동 시스템(AEB), 차선 이탈 경보 시스템(LDWS), 차선유지 지원 시스템(LKAS), 어드밴스트 스마트 크루즈 컨트롤(ASCC) 등 자율주행 자동차가 실현되기 위해선 수십 가지의 기술이 필요하다. 이는 많은 장단점에 의해 아직까지 상용화가 되지는 않고 있다.

Computational Thinking

컴퓨팅 사고력(Computational Thinking)은 컴퓨터가 문제를 해결하는 방식처럼 복잡한 문제를 단순화하고 이를 논리적, 효율적으로 해결하는 능력을 의미한다. 이는 전산적 사고로도 통용하여 쓰이며, 컴퓨팅 사고라는 용어는 Seymour Papert 교수가 처음으로 1980년에 그의 저서에서 언급되었다. 컴퓨팅 사고력을 사용해서 복잡한 대규모 문제를 알고리즘으로 해결할 수 있고, 흔히 효율성에서 상당한 개선을 달성하는데 사용되어진다. 이처럼



컴퓨터(사람이나 기계)가 효과적으로 수행할 수 있도록 문제를 정의하고 그에 대한 답을 기술하는 것이 포함된 사고 과정 일체를 일컫는데, 정답이 하나가 아니라 여러가지일 수 있는 문제(Open-ended Problem)는 다양한 변수에 기반한 포괄적이며 유의미한 해답도출이 필요할 때, 컴퓨팅 사고를 통해서 발견한 문제분해(decomposition), 자료표현(data representation), 일반화(generalization), 모형(modeling), 알고리즘으로, 전체 의사결정 과정을 분해하고, 연관된 변수와 모든 가능한 해법을 고려해서, 이를 상응하는 모수와 문제 한계를 고려하여 올바른 의사결정을 내리는 것이다. 이는 빠르게 변하고 복잡해지는 미래 사회에서 컴퓨팅적 사고력을 키우면 창의력, 문제 해결력, 사고력 등도 함께 키울 수 있고, 일상생활의 여러 문제를 효율적으로 해결할 수 있기 때문에, 모든 사람들이 갖추어야 할 능력으로 꼽히고 있다.

자동화(Automation)

자동화는 앞서, 3A에서 언급했던 것과 같이, 컴퓨터나 전자 기기를 이용하여 일 처리가 자동으로 되도록 하는 것을 일컫는다. 이는 앞서 언급한 컴퓨팅 사고력의 중요한 핵심 특징 중 하나이며, Abstraction 과 Algorithm 과 함께, 조화되어 컴퓨팅 사고력을 체계화한다. (앞서 언급했기 때문에 간략하게 설명, 추가 설명이 필요하다면 위의 3A와 Computational Thinking 참조)

Alan Turing

앨런 튜링은 1912년 6월 23일에 태어나서 1954년 6월 7일에 사망한, 수학자이자 과학자이다. 그는 컴퓨터의 아버지로 일컬어지며, 그의 가장 유명한 업적으로는 세계 최초의 몇몇 전자식 컴퓨터의 설계 및 구축의 범용 튜링머신 이론과 제2차 세계대전 중 영국 정보국(British Government Code and Cypher School)의 본부인 버킹엄셔 주에 있는 브레츨리 파크에서 한 암호해독 작업이다. 튜링은 컴퓨터공학과 정보공학의 기본 이론을 대부분 다 만든 컴퓨터 과학의 이론적 아버지였고, 이론생물학에도 큰 기여를 하여, 미국 계산기학회에서는 1966년부터 튜링의 업적을 기리는 의미로 컴퓨터 과학 분야에서 중요한 업적을 남긴 사람에게 튜링상을 수여한다. 이는 흔히 컴퓨터 과학 분야의 노벨상이라 불리는 상이다. 또한 튜링테스트를 통과한 인공지능에게는 뢰브너 상이 수여된다.

Object Detection

Object Detection은 '객체 탐지'라고 일컬으며, 이미지나 동영상에서 사람, 동물, 차량 등 의미 있는 객체(object)의 종류와 그 위치(bounding box)를 정확하게 찾기 위한 컴퓨터 비전(computer vision) 기술을 뜻한다. 이는 디지털 이미지 및 비디오에서 특정 클래스 (예 : 사람, 건물 또는 자동차)의 의미 객체를 감지하는 작업을 처리하는 기술이며, 영상에서 관심 대상을 인식하기 위해 일반적으로 검출 대상에 대한 후보 영역을 찾고 그 후보 영역에 대한 객체의 종류와 위치를 학습된 모델을 통해 예측한다. 그리고 이 과정을 위해서 영상 및 영상 내의 객체 종류(class)와 객체



위치(bounding box) 정보가 필요하고, 얼굴, 도로상의 보행자 및 차량 등의 인식에 딥 러닝(deep learning) 기반의 객체 탐지 기술 또한 많이 이용된다.

일반적으로 객체 탐지를 위한 방식은 기계 학습 기반 접근 또는 딥 러닝 기반 접근으로 분류되는데, 기계 학습 접근의 경우 SIFT, HOG 등 여러 방식 가운데 하나를 사용하여 정의한 다음 서포트 벡터 머신(SVM) 등의 기법을 사용하여 분류하는 일이 필요하다. 한편, 딥 러닝 기법은 기능을 구체적으로 정의하지 않고서도 단대단 객체 탐지를 할 수 있으며 합성곱 신경망(CNN)에 기반을 두는 것이 보통이다.



<Object Detection를 이용한 예시>

문제 해결 능력

문제 해결 능력은 문제 상황이 발생했을 경우 창조적이고 논리적인 사고를 통해 이를 적절하게 해결하는 능력이고, 이는 업무를 수행함에 있어 문제 상황이 발생했을 경우, 창조적이고 논리적인 사고를 통하여 이를 올바르게 인식하고 적절하게 해결하는 능력을 말한다. 여기서 '문제 해결'은 목표와 현상을 분석하고, 이 분석 결과를 토대로 과제를 도출하여 최적의 해결책을 찾아 실행, 평가해 가는 활동을 의미한다. 이는, 윤은영 교수님께서 기업이 가장 원하는 능력이라고 언급 하셨고, 문제해결능력은 문제 해결을 위해 사실과 의견을 구분하고 유용한 의견과 타당한 의견을 제시하는 사고력과, 문제 발생 시 사실과 대안을 확인하고 원인을 분석하며 다양한 대안을 제시하여 처리하는



문제처리능력으로 나눌 수 있다.

파이썬

파이썬은 C언어를 기반으로 한 오픈소스 고급 프로그래밍 언어로, 오픈소스 고급 프로그래밍 언어 중 하나이다. 또한 파이썬은 네덜란드 개발자 귀도 반 로섬(Guido van Rossum)이 만든 언어이며, 프로그래밍 언어에는 파이썬 뿐만 아니라 C, 자바, 자바 스크립트 등 여러 가지가 있지만 그 중에서도 간결하고 생산성 높은 프로그래밍 언어로 알려져, 가장 널리 사용되고 있는 것으로 알려져 있다. 이는 플랫폼이 독립적이기 때문에 다양한 플랫폼에서 사용 가능하고, 기본 제공되는 라이브러리가 매우 많아서 유용하다. 또한 C언어와 다르게 인터프리터식 동적 타이핑(Dynamically typed) 대화형 언어이어서, 인터프리터 형식이기 때문에 사용자가 컴파일을 하지 않고서도 작성한 프로그램을 바로 실행할 수 있을 뿐만 아니라, 한 줄 단위로 실행되기 때문에 사용자가 쉽게 결과를 확인할 수 있다.

이처럼 파이썬은 문법이 간결하고, 표현 구조가 인간의 사고 체계와 닮아 있다. 이 덕분에 초보자도 쉽게 배울 수 있고, 이러한 특징은 유지 보수와 관리도 쉽게 하도록 도우며, 외부에 풍부한 라이브러리는 다양한 용도로 확장하기에 좋다. 실제로 파이썬은 웹 개발 뿐만 아니라 데이터 분석, 머신러닝, 그래픽, 학술 연구 등 여러 분야에서도 활용되고 있으며, 생산성이 높은 장점들을 가지고 있다. 반대로 단점이라고 하면, 속도가 느리다는 평이 있으며, 모바일 앱 개발 환경에서는 사용하기 힘들다는 점이다. 또한 컴파일 시 타입 검사가 이루어지지 않아, 개발자가 실수할 여지가 조금 더 많다거나 멀티코어를 활용하기 쉽지 않다는 단점이 있다.

AI

AI는 Artificial Intelligence으로, 컴퓨터에서 인간과 같이 사고하고 생각하고 학습하고 판단하는 논리적인 방식을 사용하는 인간지능을 본 딴 고급 컴퓨터프로그램을 말한다. 즉 인간의 지능이 할 수 있는 사고·학습·모방·자기 계발 등을 컴퓨터가 할 수 있도록 연구하는 컴퓨터공학 및 정보기술 분야를 말한다.

과거의 인공지능은 확정된 환경에서 유한개의 솔루션을 탐색하는 일이었다. 인공지능은 곧 논리였고, 이에 따른 탐색이었다. 하지만 현실은 환경도 매우 불확정적이고, 솔루션도 미리 유한개로 정해져있지 않은 경우가 많아, 이를 기계학습으로 “데이터 중심의 판단”으로 풀어나갔고, 이는 초기에는 게임·바둑 등의 분야에 사용되는 정도였지만, 현재는 실생활에 응용되기 시작하면서 지능형 로봇 등 활용 분야가 비약적으로 발전하였다.

신경망, 퍼지이론, 패턴 인식, 전문가 시스템, 자연어 인식, 이미지 처리, 컴퓨터 시각, 로봇공학 등 다양한 분야가 인공지능의 일부분을 이루고 있으며, 또한 인공지능은 그 자체만으로 존재하는 것이 아니라 컴퓨터과학의 다른 분야와 직간접으로 많은 관련을 맺고 있다. 특히 현대에는 정보기술의 여러 분야에서 인공지능적 요소를 도입해 그 분야의 문제 해결에 활용하려는 시도가 활발히 이루어지고 있다.



안드로이드(Android)

안드로이드는 구글과 핸드폰 업체들이 연합하여 개발한 개방형 모바일 운영체제이다. 안드로이드는 리눅스를 기반으로 만들어졌으며, 세계 각국의 이동통신 관련 회사 연합체인 '오픈 핸드셋 얼라이언스(OHA ; Open Handset Alliance)'가 2007년 11월에 공개하였다. 실질적으로는 구글(Google)사가 안드로이드사를 인수하여 개발 하여, '구글 안드로이드'라고도 한다. 안드로이드는 리눅스(Linux) 2.6 커널을 기반으로 강력한 운영체제(OS ; operating system)와 포괄적 라이브러리 세트, 풍부한 멀티미디어 사용자 인터페이스, 폰 애플리케이션 등을 제공한다. 컴퓨터에서 소프트웨어와 하드웨어를 제어하는 운영체제인 '윈도'에 비유할 수 있는데, 휴대폰에 안드로이드를 탑재하여 인터넷과 메신저 등을 이용할 수 있으며, 휴대폰뿐 아니라 다양한 정보 가전 기기에 적용할 수 있는 연동성도 갖추고 있다. 안드로이드는 완전 개방형 플랫폼이라는 점에서, 다른 휴대폰 운영체제인 애플의 'iOS', 마이크로소프트의 '윈도 모바일', 노키아의 '심비안'등과 차별화 된다. 이는 종전에는 휴대폰 제조업체와 서비스업체마다 운영체제가 달라 개별적으로 응용프로그램을 만들어야 했기 때문인데, 안드로이드는 기반 기술인 '소스 코드'를 모두 공개함으로써 누구라도 이를 이용하여 소프트웨어와 기기를 만들어 판매할 수 있도록 하였다. 하여 개발자들은 이를 확장, 대체 또는 재사용하여 사용자들에게 풍부하고 통합된 모바일 서비스를 제공할 수 있게 되고, 사용량이 크게 늘게 되었다.

깃허브(GitHub)

깃허브는 컴퓨터 프로그램 소스를 공유하고 협업하여 개발할 수 있는 버전 관리 시스템인 깃(Git)에 프로젝트 관리 지원 기능을 확장하여 제공하는 웹 호스팅 서비스이다. 2008년 미국 깃허브사(GitHub Inc)에서 시작되었고, 사용자에게 무료로 계정과 저장소를 제공하며, 분산형 버전 관리 서비스로 서버 장애 시 데이터 복원력이 뛰어나서 현재 전 세계에서 오픈 소스 프로젝트 관리를 위해 가장 많이 사용되는 웹 호스팅 서비스 중 하나이다. 깃(Git)은 2005년 리눅스 제작자인 리누스 토발즈(Linus Torvalds)가 오픈 소스 리눅스(Linux) 커널 개발의 효율성을 높이기 위해서 개발하였으며, 많은 개발자들이 소프트웨어 소스 코드를 공유하고 협력하면서 개발할 수 있도록 지원하는 분산형 버전 관리 시스템(DVCS: Distributed Version Control System)으로, 코드 수정 권한, 버전 추적 등 소프트웨어 개발에 필요한 관리 기능을 제공한다. DVCS에서는 공동 버전 시스템(CVS)과 달리, 클라이언트가 저장소를 통째로 복제하여 사용하기 때문에 서버에 문제가 발생해도 클라이언트는 복제된 저장소를 다시 서버에 복사하여 서버 내 데이터를 복원할 수 있어 많이 사용된다. 이는 깃허브의 조사에 따르면, 2018년 6월 기준으로 2,800만 명 사용자와 5,700만개의 코드 저장소로 파악되어 많은 사람들이 사용 중임을 알 수 있다.

JAVA

JAVA는 객체지향프로그래밍 언어로서 C/C++에 비해 간략하고 쉬우며 네트워크 기능의 구현이 용이하기 때문에, 인터넷 환경에서 가장 활발히 사용되는 프로그래밍 언어이다. 1991년 제임스 고슬링을 포함한 썬 마이크로시스템즈의 연구원들에 의해 인터랙티브 TV(Interactive TV)용 프로그램 작성을 위해 처음 개발되었고, 자바 언어는

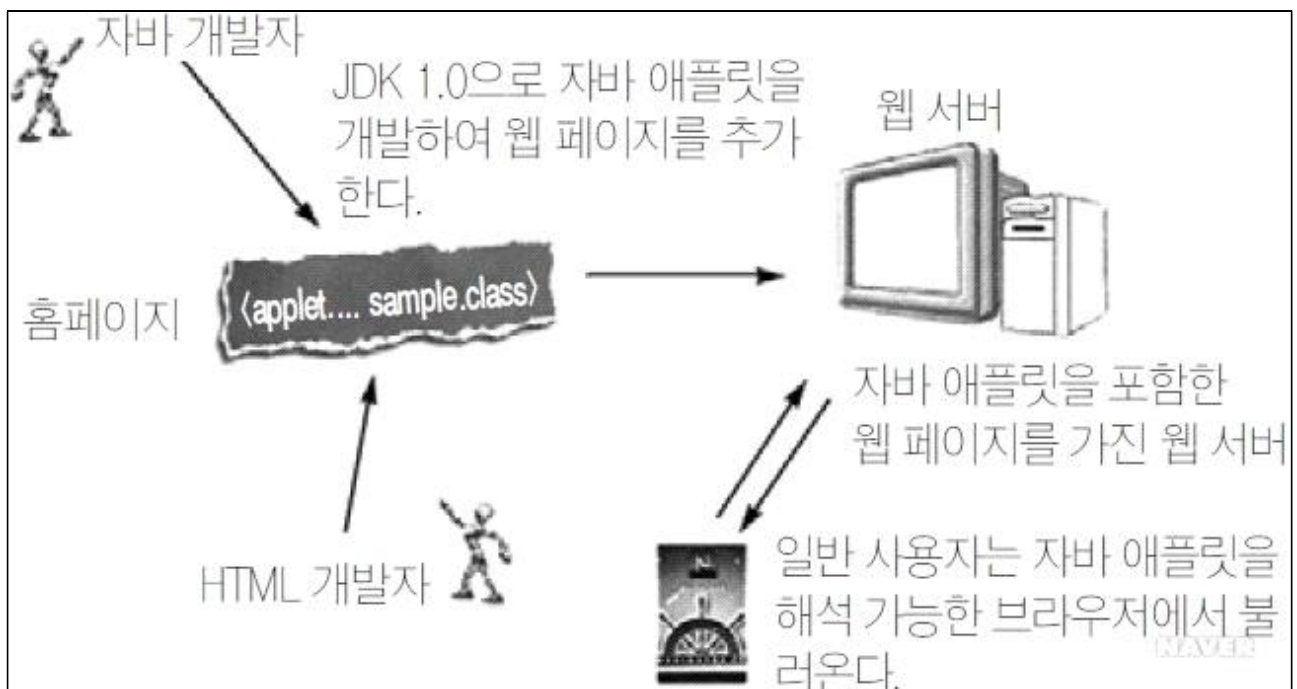


객체지향프로그래밍 언어로서, 시스템 및 어플리케이션 개발자들에게 익숙하도록 C/C++과 유사한 문법을 가지고 있는 반면 C++에 비해 단순하고 효율성을 높이기 위해 여러 기능이 추가되어 있다.

자바의 특징을 설명하자면, 간단하고, C++를 기반으로 개발되었음에도 C++에서 혼란을 일으키는 부분을 제거하여, C++에서 문제가 되는 기억 장소 경영 문제를 자동 쓰레기 회수와 할당(auto garbage collection)으로 극복 되어 있으며, 객체 지향 언어(object-oriented)이고, 보안에 강하며, 아키텍처는 중립적(architecture neutral)이고, 이식성이 높은 것을 꼽을 수 있다. JAVA Script와는 둘 다 객체 지향적 언어이나, 자바 스크립트에는 존재하지 않는 상속성이나 클래스가 존재한다는 것과, 항상 변수형을 선언해야 하는 스트롱 타입(strong type)임과, 정적인 연결(static binding)을 취한다는 것, 두 언어 모두 안전하지만, 자바 스크립트는 HTML 코드에 직접 끼워져 있어서 누구든지 볼 수 있으나, 자바는 다른 사람이 그 코드를 볼 수 없음에 차이가 있다.



<JAVA 언어의 역사>



<JAVA 애플릿이 웹 상에서 구현되는 원리>