Ch01_지식기반 지능형 시스템 개론

이 장에서 다룰 내용

- 01_지능형 기계가 할 수 있는 것
- 02_'암흑기'에서 지식기반 시스템에 이르는 AI의 역사
- 03_요약

01_지능형 기계가 할 수 있는 것

❖ '지능(intelligence)'이라는 단어의 의미

1. 인간의 지능은 무언가를 이해하고 배우는 능력이다.
2. 지능은 본능적 혹은 자동적으로 무언가를 하는 대신 생각하고 이해하는 능력이다.

_Essential English Dictionary, Collins, London, 1990

- **첫 번째 정의에 따르면, 지능은 인간에게 있는 성질.**
- **두 번째 정의에 따르면, 생각하고 이해하는 능력을 갖춘 것이 사람인지 사물인지 명확하지 않음.**

❖ '생각'이라는 단어의 의미

생각 (thinking) 이란 문제를 고려하거나 아이디어를 만들기 위해 두뇌를 사용하는 행위다.

_Essential English Dictionary, Collins, London, 1990

- **생각을 하려면 인간이나 사물에 두뇌**
- **즉, 지능이란 '문제를 풀고 결정을 내리기 위해 배우고 이해하는 능력'으로 정의할 수 있음.**

01_지능형 기계가 할 수 있는 것

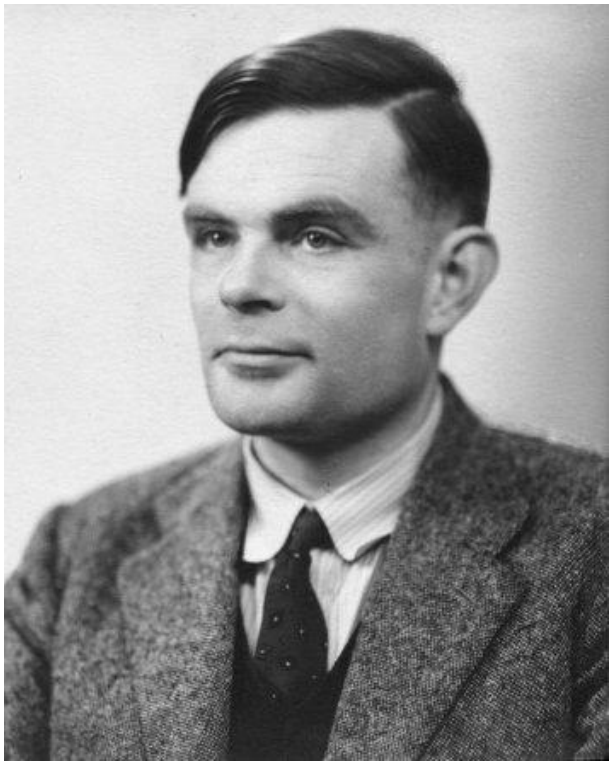
❖ 인공지능 (AI, artificial intelligence)

- 인공지능의 탄생
  - 컴퓨터가 지능적일 수 있는지, 즉 기계가 생각을 할 수 있는지에 대한 질문은 인공지능의 '암흑기(dark ages)'에서 시작.
  - 인공지능이라는 학문이 추구하는 바는 인간이 머리를 써서 해야 할 일을 기계가 하도록 만드는 것임(Boden,1977).
  - 인공지능에서는 '기계가 생각을 할 수 있는지'에 대한 답이 매우 중요.

- 인공지능 탄생의 공헌자 : 알란 튜링(Alan Turing)
  - 기계지능에 관한 초기의 의미 있는 논문 중 하나인 'Computing machinery and intelligence'을 50년도 이전에 작성.
  - 튜링의 접근 방식은 오래전에 나온 것이긴 하지만, 기나긴 테스트를 통과하여 보편적인 것이 됨.
  - 튜링은 기계와 생각의 정의를 내리지 못했고, 튜링 모방 게임(Turing imitation game)을 만듦으로써 의미론적인 논증을 피함.

- 튜링 모방 게임 (파티에서 흔히 하는 게임에서 유래)
  - 튜링은 컴퓨터의 지능적 행동을 인지 작업에서 인간 수준의 성능을 낼 수 있는 능력으로 정의함. 즉, 질문자가 질문에 대한 답변을 보고 답변자가 인간인지 기계인지 구별하지 못한다면 컴퓨터는 테스트를 통과한 것임.
  - 튜링이 제안한 튜링 모방 게임은 두 단계로 나뉨.

참고 – 앨런 튜링

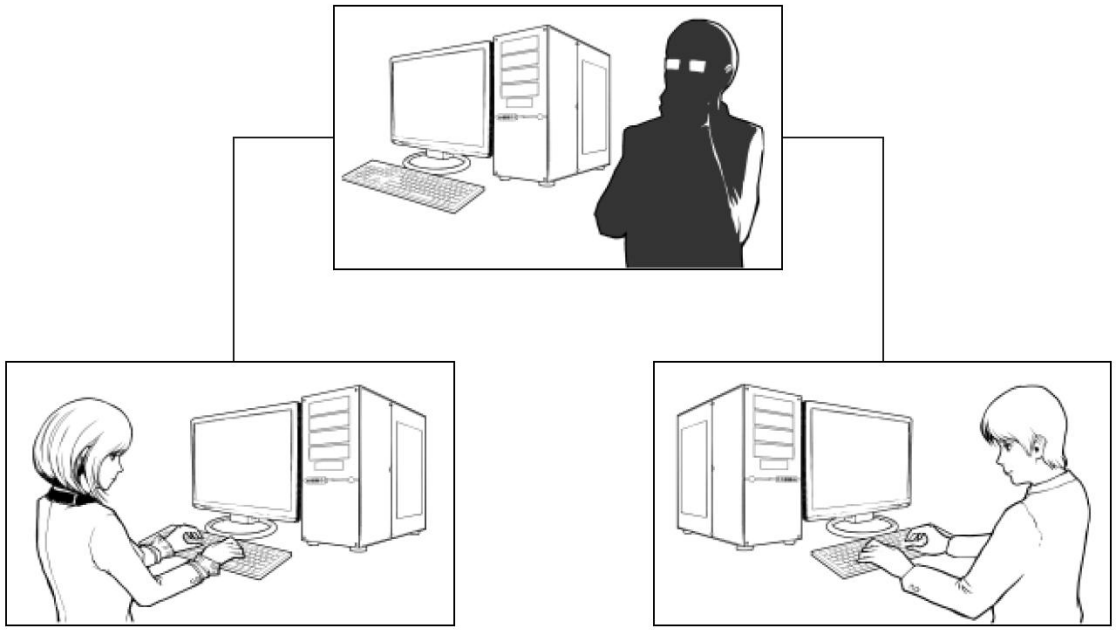


- 영국의 수학자, 암호학자, 논리학자
- 컴퓨터 과학에 지대한 공헌을 했기 때문에 '컴퓨터 과학의 아버지'
- 튜링 테스트와 튜링 기계의 고안

01_지능형 기계가 할 수 있는 것

❖ 인공지능

- **튜링 모방 게임의 첫 번째 단계**
  - 질문자, 남자, 여자 한 명이 각각 독립된 방에 있고, 원격 터미널과 같은 중립 매체를 통해서만 통신할 수 있음.
  - 질문자의 목적은 상대방에게 질문하면서 누가 남자고 누가 여자인지 알아내야 함.
  - 남자는 자신이 여자라고 질문자를 속이고 여자는 질문자에게 자신이 여자라는 확신을 심어주는 것이 게임의 규칙임.

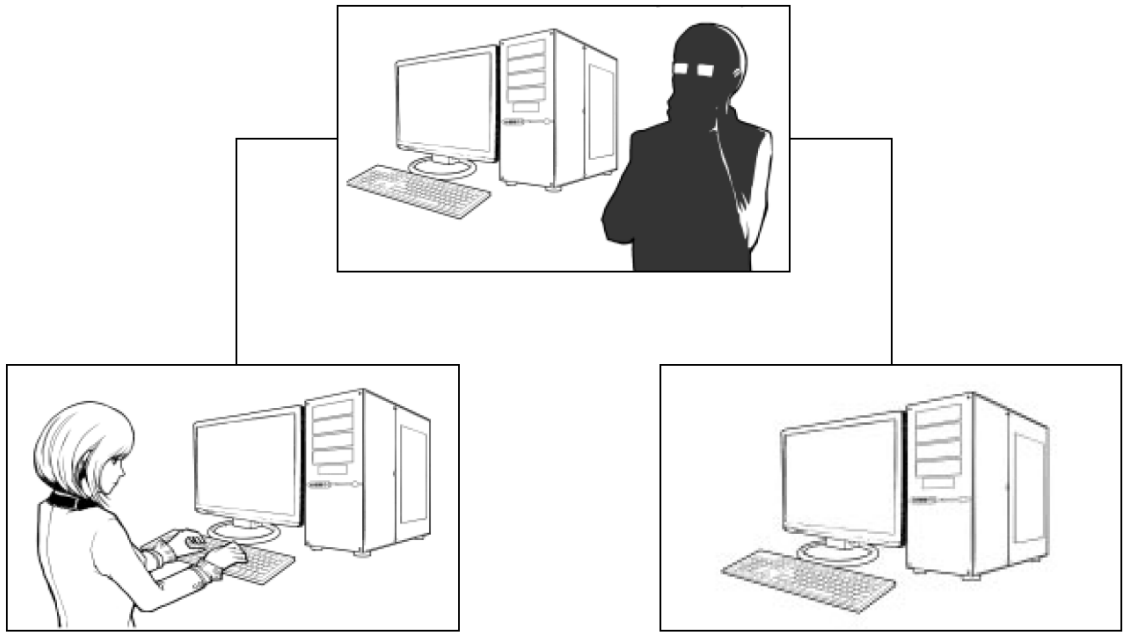


[그림 1-1] 튜링 모방 게임 : 단계 1

01_지능형 기계가 할 수 있는 것

❖ 인공지능

- **튜링 모방 게임의 두 번째 단계**
  - 두 번째 단계에서는 남자 대신 질문자를 속이도록 프로그래밍한 컴퓨터를 둠. 사람이 하는 대로 실수도 하고 애매한 답변을 하는 것까지 프로그래밍함.
  - 만약 컴퓨터가 남자가 했던 것만큼 질문자를 자주 속일 수 있다면 이 컴퓨터는 지능 행동 테스트를 통과한 것임.



[그림 1-2] 튜링 모방 게임 : 단계 2

01_지능형 기계가 할 수 있는 것

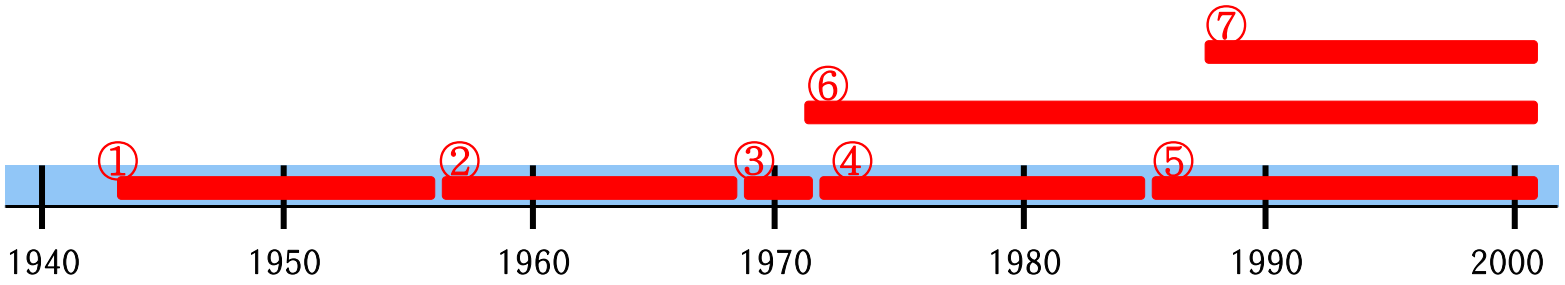
❖ 인공지능

- 튜링 모방 게임의 특징
  - 사람의 물리적인 특성은 지능을 판단할 때 중요한 요소가 아님.
  - 튜링 테스트에서 질문자는 컴퓨터를 보거나 만지거나 듣지 못하므로 외양과 소리에 영향을 받지 않음.
  - 질문자는 컴퓨터가 인간보다 빠르게 올바른 해를 제시할 것을 기대하면서 인간과 기계 모두에게 복잡한 수학 계산을 하게 할지 모름. 그러므로 컴퓨터는 언제 실수를 할지, 언제 답변을 지연할지를 알아야 함.
  - 질문자는 인간의 감정적인 본성을 찾으려 컴퓨터와 인간에게 단편소설, 시, 심지어 그림까지 살펴보게 할 수 있음. 이때 컴퓨터는 작품을 감상하는 인간의 감정 상태를 흉내낼 수 있어야 함.

- 튜링 모방 게임의 의의
  - 튜닝 테스트는 터미널을 통해 인간과 기계 사이의 통신을 유지하면서 지능에 관한 객관적이고 표준적인 시각을 제공. 이는 지능이 인간의 본성이란 논쟁을 피하고 인간에 대한 편견도 모두 없앰.

01_지능형 기계가 할 수 있는 것

❖ AI의 역사



① 암흑기: AI의 탄생(1943년 ~ 1956년)
② AI의 융성 : 큰 기대의 시기(1956년 ~ 1960)
③ 이행되지 않은 약속 : 현실의 직면(1960년대 후반~1970년대 초반)
④ 전문가 시스템의 기술 : 성공의 열쇠(1970년대 초반~1980년대 중반)
⑤ 기계가 학습하는 법 : 신경망의 재탄생(1980년대 중반~)
⑥ 진화 연산 : 탐색하면서 배우기(1970년대 초반~)
⑦ 지식 공학의 새로운 시대 : 단어로 계산하기(1980년대 후반~)

02_'암흑기 '에서 지식기반 시스템에 이르는 AI의 역사

❖ **암흑기: AI의 탄생(1943년 ~ 1956년)**

❖ **'암흑기'의 공헌자**

- **워렌 맥클록** (Warren McCulloch)**과 월터 피츠**(Walter Pitts)
  - AI **분야로 인정받은 최초 연구를 소개.** (1943년)
  - AI**의 첫 번째 주요 논문 발표 - 뇌의 뉴런 모델**(model of neurons of brain)
  - **인공 신경망**(artificial neural networks) **모델을 제안.** (1943년)
  - **인공 신경망에서 각 뉴런은 이진 상태,** 즉 on **또는** off **상태에 있다고 가정함.**
  - **계산이 가능한 함수라면 뉴런이 연결된 모든 망에서 계산할 수 있음을 증명.**

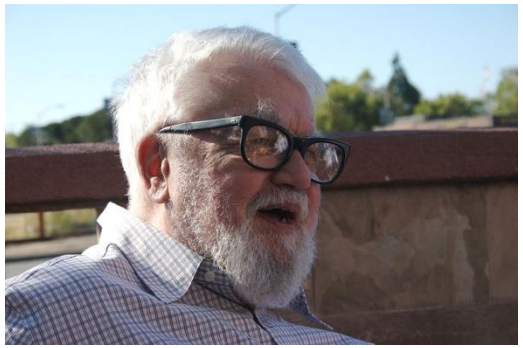
- **클라우드 섀넌**(Claude Shannon)
  - **일반 체스게임이** 10^{120}**번의 이동횟수를 포함한다는 점을 지적하는 체스게임 기계에 관한 논문 발표.** (1950년)
  - **해를 찾을 때 휴리스틱을 사용해야 한다는 사실을 증명.**

- **존 맥카시**(John McCarthy)
  - **기계 지능, 인공신경망, 오토마타 이론에 관심 있는 연구자를 모아** Dartmouth **대학교에서 여름 워크샵 열도록 도움.** (1956년)
  - Dartmouth **워크샵에서 인공지능이라는 새로운 과학 분야가 탄생.**

02_'암흑기 '에서 지식기반 시스템에 이르는 AI의 역사

❖ AI의 융성 : 큰 기대의 시기(1956년 ~ 1960)

- 초기 AI는 '엄청난 열광', '근사한 아이디어', '매우 제한된 성공'으로 특징지을 수 있음.
- 반복적인 수학 계산을 위해 컴퓨터를 도입했으나, 당시 AI 연구자는 컴퓨터가 그보다 훨씬 많은 일을 할 수 있음을 보임.



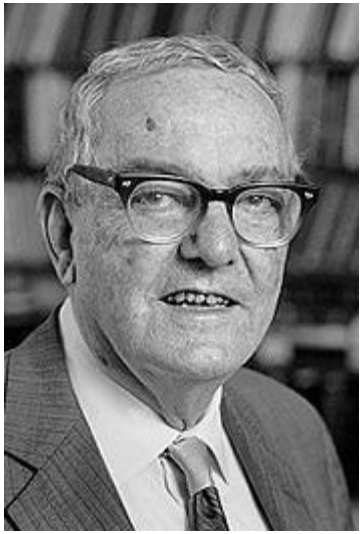
❖ 'AI의 융성'의 공헌자

- 존 맥카시 (John McCarthy)
  - 인공 지능이라는 용어를 처음 고안함
  - LISP를 정의 함.
  - 「Programs with Common Sense 」 논문에서 세상의 일반적인 문제에 대한 해를 찾는 전문가 의견 청구자(Advice Taker)라는 프로그램을 제안함.
  - 단순 공리에 기초한 이 프로그램으로 공항에 가는 계획을 생성하는 방법을 보여줌.
  - 다시 작성하지 않고도 다른 분야에서 새로운 공리를 사용할 수 있도록 설계.
  - 전문가 의견 청구자는 지식 표현 및 추론 중심의 원칙을 통합하는 첫 번째 완전한 지식기반 시스템.

02_'암흑기 '에서 지식기반 시스템에 이르는 AI의 역사

❖ 'AI의 융성'의 공헌자

- 마빈 민스키 (Minsky)
  - 형식 논리에 초점을 두었던 맥카시와 달리 민스키는 지식 표현과 추론에 관한 반논리적인 사고 방식을 발전시킴.
  - 프레임(frame) 이론은 지식 공학에 많은 공헌을 함. (1975)
- 프랭크 로젠블랫(Frank Rosenblatt)
  - 학습 알고리즘이 퍼셉트론(perceptron)의 연결 강도를 조정할 수 있다는 것을 보임.
  - 퍼셉트론 수렴 이론(perceptron convergence theorem)을 증명.
- 앨런 뉴웰(Allen Newell)과 허버트 사이먼(Herbert Simon)
  - 인간의 문제해결 방식을 모방하는 범용 프로그램 GPS(General Problem Solver)을 개발.
  - GPS는 데이터와 문제해결 기법을 구분한 첫 번째 시도임.
  - 방법-결과 분석(means-ends analysis)이라는 기법에 기반을 두고 있음.



❖ GPS프로젝트

- 뉴웰과 사이먼이 개발(1961, 1972)
- GPS는 데이터와 문제해결 기법을 구분한 첫 번째 시도임.

02_'암흑기 '에서 지식기반 시스템에 이르는 AI의 역사

❖ GPS(General Problem Solver) 프로젝트

- 방법-결과 분석(means-ends analysis)이라는 기법에 기반을 둠.
- GPS는 복잡한 문제를 푸는 데 실패함.
  - 형식 논리에 기반을 둔 프로그램이 사용 가능한 연산자를 무한정 생산함.
  - 현실 세계 문제를 푸는 데 필요한 시간과 메모리의 양 때문에 진행이 중단됨.

❖ 'AI의 융성'의 의의

- AI 연구자들은 광범위한 문제를 풀기 위한 일반 방법(general methods)을 만들어서 복잡한 사고 과정을 모의함.
  - 약한 방법(weak methods)이라고도 함.
  - 문제 영역에 적합하지 않은 정보를 사용했고 이는 개발한 프로그램의 성능을 저하시켰음.

- AI 분야에 매력을 느낀 위대한 과학자들이 지식 표현, 학습 알고리즘, 신경 컴퓨팅, 단어계산과 같은 분야에 새롭고 중요한 아이디어를 제안함.

02_'암흑기 '에서 지식기반 시스템에 이르는 AI의 역사

❖ 이행되지 않은 약속 : 현실의 직면(1960년대 후반~1970년대 초반)

- AI 연구자들의 낙관과 실패
  - AI 연구자들은 1950년대 중반부터 1980년대까지는 인간 규모의 기반지식을 가진 범용 지능형 기계를 만들고, 2000년에는 인간의 지능을 넘어서게 하겠다고 약속함. 그러나
  - 1970년에 이르러 그런 주장은 너무 낙관적이었다는 사실을 깨달음.

- AI의 프로그램의 한계
  - 몇몇 AI 프로그램이 현실 문제를 간단하게 만든 장난감 문제 한두 개에서 일정 수준의 기계 지능을 보여줌.
  - 하지만 대부분의 AI 프로젝트는 작업의 범위를 넓히거나 더 어려운 실 세계 문제를 다루지 못함.

❖ 1960년대 후반에 AI에 관한 주된 어려움

- AI 연구자는 광범위한 문제를 해결할 일반적인 방법을 개발하고 있었기 때문에 초기 프로그램에는 특정 문제 영역에 관한 지식을 거의 포함하지 못함.
  - 문제를 해결하기 위해 프로그램은 해를 발견할 때까지 작은 단계들을 여러 조합으로 만들어보는 탐색 전략을 적용.
  - '장난감 문제'과 같은 작은 문제에서 잘 동작. 그러나 어렵고 복잡한 큰 문제에서 잘 동작하지 않음.

02_'암흑기 '에서 지식기반 시스템에 이르는 AI의 역사

❖ 1960년대 후반에 AI에 관한 주된 어려움

- AI로 풀려고 한 많은 문제는 범위가 넓고 풀기도 어려움
  - 초기 AI 작업은 대개 기계 번역.
  - 올바른 단어를 선택하려면 주제를 먼저 이해해야 한다는 사실을 알게 됨. 그러나 이는 너무 어려운 과정이었음.

- AI에 대한 지원 중단
  - 1966년에 미국 정부의 지원을 받았던 AI를 이용한 모든 번역 프로젝트 취소.
  - 1971년에 영국 정부도 AI 연구에 대한 지원 중단.

02_'암흑기 '에서 지식기반 시스템에 이르는 AI의 역사

❖ 전문가 시스템의 기술 : 성공의 열쇠(1970년대 초반~1980년대 중반)

- 지능형 기계에 대한 문제 영역을 충분히 제한해야 한다는 사실을 깨달음.
  - 이전의 AI 연구자는 일반적이며 인간의 문제 풀이 방법을 흉내 낸 똑똑한 탐색 알고리즘과 추론 기법을 만들 수 있다고 믿음(약한 방법).
  - 하지만, 약한 방법은 실패했고 이를 통해 연구자들은 전문지식이 필요한 전형적인 영역으로 문제를 제한하고 큰 추론단계로 해결해야 실용적인 결과를 얻을 수 있다는 것을 깨달음.

❖ '전문가 시스템의 기술' 시대의 주요 프로젝트

- DENDRAL 프로젝트
  - 질량 스펙트럼을 결정하는 시스템.
  - 파이겐바움, 뷰캐넌, 레더버그가 만든 AI 시스템.
  - 인간 전문가의 기술과 경험, 의견을 이용하여 문제를 해결.
  - 전문가 시스템(expert systems)이라고 함.

- DENDRAL 프로젝트의 중요성
  - AI에서 핵심적인 '패러다임 전환' 을 만듬. 범용으로 쓰이는 (지식이 빈약한) 약한 방법에서 특정 분야의 (지식 집중적인) 방법으로 전환됨.

02_'암흑기 '에서 지식기반 시스템에 이르는 AI의 역사

- DENDRAL 프로젝트의 중요성
  - 인간 전문가에게서 이끌어낸 양질의 특수 규칙 형태로 휴리스틱을 사용하여 컴퓨터가 제한된 문제 영역에서 전문가와 같을 수 있음을 보였음.
  - 전문가 시스템을 만드는 새로운 방법론의 기초 개념인 지식 공학(knowledge engineering)을 만듬.
  - 프로젝트에는 전문가의 '노하우'를 규칙으로 얻고 분석하여 표현하는 기법을 담았음.

- MYCIN 프로젝트
  - 전염성 혈액 질환을 진단하는 규칙기반 전문가 시스템.
  - MYCIN은 관련 분야에서 인간 전문가와 동일한 수준으로 수행할 수 있었고, 경험이 부족한 의사보다 오히려 수준이 높았음.
  - 규칙의 형태로 통합된 지식은 추론 메커니즘과 분리함. 시스템 개발자는 일부 규칙을 추가하거나 삭제함으로써 시스템 내의 지식을 쉽게 다룰 수 있음.
  - MYCIN에 통합된 규칙은 지식과 관련된 불확실성을 반영함.

- ROSPECTOR 프로젝트
  - 광물탐사 전문가 시스템.
  - 프로젝트는 1974년~1983년 동안 수행되었으며, 전문가 아홉 명이 자신의 지식과 의견을 제공.
  - 지식을 표현하기 위해 규칙과 의미망(semantic network)을 결합한 구조를 사용.

02_'암흑기 '에서 지식기반 시스템에 이르는 AI의 역사

- 프로젝트의 특징
  - 1970년대 후반에 전문가 시스템을 성공적으로 응용한 사례가 증가하면서 연구소에 머물러있던 AI 기술이 상업적인 환경으로 성공적으로 전이함.
  - 이 기간 동안 대부분의 전문가 시스템은 강력한 워크스테이션에 기반을 둔 LISP, PROLOG, OPS와 같은 특별한 AI 언어로 개발됨.
  - 비싼 하드웨어와 복잡한 프로그래밍 언어가 있어야 한다는 조건 때문에 전문가 시스템 개발은 스탠포드 대학교, MIT, 스탠포드 연구소, 카네기멜론 대학교의 몇몇 연구 그룹에서 이루어짐.
  - 1980년대 개인용 컴퓨터(PC)와 사용하기 쉬운 전문가 시스템 개발 툴(shell)이 등장하고 나서야 모든 분야의 일반 연구자와 공학자가 전문가 시스템을 개발할 기회를 얻음.

❖ 전문가 시스템은 어떤 분야에서도 성공할 수 있을까?

- 지식이 서로 다른 분야에서 상당히 많은 전문가 시스템을 성공적으로 개발하고 구현했지만, 이 기술력을 높이 평가하는 일은 아직 시기상조임.
- 전문가 시스템의 제약 사항
  - 전문가 시스템의 사용은 매우 한정된 전문적 기술분야로 제한됨.
  - 한정된 분야 때문에 전문가 시스템은 사용자가 원하는 만큼 신뢰할 수 없고 유연하지도 못함. 더욱이, 전문가 시스템은 분야의 경계를 인식하기 어려움.

02_'암흑기 '에서 지식기반 시스템에 이르는 AI의 역사

- 전문가 시스템의 제약 사항
  - 전문가 시스템은 제한적으로만 설명할 수 있음.
  - 해에 도달하기 위해 적용했던 일련의 규칙을 보여줄 수는 있지만, 누적된 휴리스틱 지식으로 문제 영역을 더 자세히 이해하게 할 수는 없음.
  - 전문가 시스템은 결과를 검증하고 유효성을 입증하기 어려움.
  - 1세대 전문가 시스템은 자신의 경험을 통해 배울 수 있는 능력이 없음.

02_'암흑기 '에서 지식기반 시스템에 이르는 AI의 역사

❖ 기계가 학습하는 법 : 신경망의 재탄생(1980년대 중반~)

- 1980년대 중반 연구자, 공학자, 전문가는 전문가 시스템을 만들기 위해서는 추론 시스템 혹은 전문가 시스템 틀을 구입해서 그 안에 규칙을 넣는 것보다 훨씬 더 많은 작업을 해야 한다는 사실을 알게 됨.
- AI 연구자들은 새로운 시각으로 신경망을 봄.

❖ 신경망의 부활

- 1960년대 후반까지 신경 컴퓨팅에 필요한 대부분의 기본 아이디어와 개념은 이미 공식화되었지만, 1980년대 중반에 와서야 겨우 해법이 나타남.
- 신경망 분야는 1980년대에 컴퓨터 기술이 발전하고, 신경 과학이 진보하면서 뇌 같은 정보 처리과정이 필요하여 드라마 같이 부활함.
- '신경망 부활'의 공헌자
  - 그로스버그가 자기조직의 '적응형 공진' 이론을 세워, 신경망에 대한 기초를 제공.
  - 홉필드는 피드백이 있는 신경망인 '홉필드 신경망'을 제안.
  - 코호넨이 '자기조직 맵'에 관한 논문을 출판.
  - 브라이슨과 호가 '역전파 학습 알고리즘'을 재발견함.

02_'암흑기 '에서 지식기반 시스템에 이르는 AI의 역사

❖ 진화 연산 : 탐색하면서 배우기(1970년대 초반~)

- AI의 진화론적 관점은 자연 선택과 유전학 계산 모델에 기반을 둠.
- 진화 연산(evolutionarycomputation)은 해 집단 모의, 성능 평가, 새로운 해 집단 생성 과정을 반복함.
- 진화 연산은 유전 알고리즘(genetic algorithm), 진화전략 (evolutionary strategies), 유전 프로그래밍(genetic programming)을 결합함.

❖ 진화 연산 알고리즘

- 유전 알고리즘
  - 1970년대 초에 존 홀랜드(John Holland)가 도입.
  - 홀랜드가 선택, 교차, 변이와 같은 유전적 연산을 사용하여 인공적인 '염색체'를 다루는 알고리즘을 개발함.
  - 스키마 정리라는 믿을 수 있는 이론에 기반을 둠.

- 진화 전략
  - 1960년대 초에 홀랜드의 유전 알고리즘과는 별도로 베를린 공과 대학교에 재학중인 잉고 레켄베르그(Ingo Rechenberg)와 한스-폴 쉬베펠(Hans-Paul Schwefel)은 새로운 최적화 방법으로 진화 전략을 제안.

02_'암흑기 '에서 지식기반 시스템에 이르는 AI의 역사

❖ 진화 연산 알고리즘

- 진화 전략
  - 돌연변이가 발생하는 것처럼 매개변수에 임의의 변화를 주는 방법을 제안함.
  - 진화 전략 접근법은 공학자의 직관을 대신할 수 있다. 진화전략은 몬테카를로 (Monte Carlo) 탐색과 유사하게 수치적인 최적화 절차를 수행함.

- 유전 프로그래밍
  - 프로그램을 만들기 위해 학습하는 유전 모델을 응용한 예.
  - 유전 프로그래밍은 어떤 문제의 코드 표현을 진화시키는 것이 아니라 문제를 해결하는 컴퓨터 코드를 진화시키는 것을 목표로 함.
  - 유전 프로그래밍은 문제에 대한 답으로 컴퓨터 프로그램을 생성함.

❖ 진화 연산의 의의

- 유전 알고리즘은 이전에 풀리지 않았던 고도로 복잡한 비선형 탐색 및 최적화 문제에 대해 올바른 해를 제시함.
- 유전 프로그래밍은 구체적으로 프로그래밍하지 않고도 컴퓨터가 문제를 풀게 하는, 컴퓨터 과학의 주요 도전과제에 대한 해법을 제시.
- 유전 알고리즘, 진화 전략, 유전 프로그래밍은 AI에서 빠르게 성장하고 있는 영역이며 발전 가능성이 무한함.

02_'암흑기 '에서 지식기반 시스템에 이르는 AI의 역사

❖ 지식 공학의 새로운 시대 : 단어로 계산하기(1980년대 후반~)

❖ '단어 계산'의 탄생 배경

- 신경망 기술의 한계
  - 신경망 기술은 심벌 추론 기반 시스템보다 실세계와 자연스럽게 상호 작용을 함.
  - 신경망은 배우고, 문제 환경의 변화에 적응하고, 규칙이 알려지지 않은 상황에서 패턴을 찾고, 분명하지 않거나 불완전한 정보를 다룸.
  - 신경망은 설명 능력이 부족하고 블랙박스처럼 동작과정을 알 수 없음.
  - 현재의 기술로 신경망을 훈련하기까지 시간이 너무 오래 걸리고, 빈번한 재훈련은 심각한 문제를 유발할 수 있음.

- 전문가 시스템의 한계
  - 고전적인 전문가 시스템은 정확하게 입력하고 논리적으로 출력하는 닫힌 시스템에 적합.
  - 규칙의 형태로 전문가 지식을 사용하고, 필요하다면 특정 사실을 확립하기 위해 사용자와 대화할 수도 있음.
  - 전문가 시스템의 주요 결점은 인간 전문가가 자신의 지식을 규칙의 형태로 항상 표현할 수는 없으며, 자신의 추론 과정을 항상 설명할 수 없다는 점임. 이는 전문가 시스템이 필요한 지식을 축적하는 것을 방해하여, 결과적으로 실패하게 만듬.

02_'암흑기 '에서 지식기반 시스템에 이르는 AI의 역사

❖ '단어 계산'의 탄생 배경

- '단어 계산'의 탄생
  - 신경망 기술과 전문가 시스템을 보완하기 위해 탄생
  - '단어 계산'은 불완전하다면 지식을 다루면서, 결론에 대한 설명.
  - 모호하고 부정확하며 불확실한 지식과 데이터를 다루는 중요한 기술 중 하나는 퍼지 논리(fuzzy logic)임.

❖ 퍼지 논리

- 퍼지 논리의 탄생 배경
  - 고전적인 전문가 시스템에서 부정확함을 다루는 대부분의 방법은 확률 개념에 기반을 둠.
  - 전문가는 보통 확률 값으로 생각하지 않고 '종종', '일반적으로', '가끔', '자주', '드물게'와 같은 용어로 생각함.
  - 어렵고 복잡한 문제에 대해 전문가의 이해를 정확히 반영하는 형태로 인간의 지식을 코드화하여 적용하며, 전통적인 전문가 시스템의 계산 병목 현상(computational bottleneck)을 극복하기 위해 제안됨.

- 퍼지 논리의 특징
  - 퍼지 논리는 단어의 의미를 얻고 추론과 결정을 내리기 위해 퍼지 값을 사용함.
  - 언어 변수의 값은 숫자라기보다는 단어임.

❖ 퍼지 논리

- **퍼지 논리의 특징**
  - **전문가 시스템과 유사하게 퍼지 시스템은 인간의 지식을 통합하기 위해 IF-THEN 규칙을 사용하지만, 이 규칙은 다음과 같이 분명하지 않음.**

IF 속도가 빠르다	THEN 정지거리는 길다
IF 속도가 느리다	THEN 정지거리는 짧다

- **퍼지 논리의 공헌자 : 로트피 자데(Lotfi Zadeh)**
  - **1965년에 퍼지 논리 혹은 퍼지 집합론(fuzzy set theory)을 소개함.**
- **퍼지 논리의 응용 사례**
  - **1987년 이래로 일본인이 설계한 식기세척기, 세탁기, 에어컨, 텔레비전, 복사기, 심지어 자동차에 성공적으로 사용됨.**
  - **대부분의 퍼지 논리 응용 사례는 제어 공학 분야에서 등장함. 하지만 퍼지 제어 시스템은 지식을 표현하는 퍼지 논리 능력의 일부만 사용함.**

❖ 지식기반 결정지원 시스템에서의 퍼지 논리 모델 응용 이점

- **향상된 계산 능력**
  - 퍼지 시스템은 규칙 개수가 적어 전문가 시스템보다 더 빠르게 계산을 수행함.
  - 퍼지 전문가 시스템은 규칙을 더 강력하게 만들면서 규칙을 통합함.

- **향상된 인지 모델링**
  - 퍼지 시스템은 복잡한 문제에 대한 지식을 인코딩 할 때 전문가가 생각하는 방식을 반영함.
  - 퍼지 전문가 시스템은 전문가가 마음속으로 생각하는 방식과 훨씬 가까운 의견을 얻고, 부정확한 정보를 모델링하여 문제에 대한 인지 모델링을 개선함. (기존형식으로 규칙을 만들려면 전문가 의견을 조각내어 용어의 경계를 정의해야 함. 단편화는 고도로 복잡한 문제를 다룰 때 품질 낮은 보통의 전문가 시스템으로 만듬)

- **여러 전문가 의견을 표현하는 능력**
  - 전문가 시스템은 명확히 정의된 전문가 의견이 있는 매우 한정된 분야에서 사용함. 이 때문에 시스템의 성능은 올바른 전문가의 선택에 의존함.
  - 퍼지 전문가 시스템은 반대 의견이 있을 때 여러 전문가가 의견을 표현할 수 있도록 도울 수 있음.

❖ 퍼지 시스템의 한계

- 퍼지 시스템이 전문가 지식을 자연스럽게 표현한다고 해도 여전히 전문가에게 얻은 규칙에 의존함.
- 전문가의 지식에 따라 시스템 성능이 달라짐.
- 어떤 전문가는 매우 똑똑한 퍼지 규칙을 제공할 수 있지만, 어떤 전문가는 단지 추측만 하고 심지어 잘못된 규칙을 제시할 수 도 있음. 그러므로 모든 규칙을 검사하고 조율해야 함.

03_요약

❖ 지식기반 지능형 시스템 개론 요약

- 지능은 배우고 이해하며 문제를 풀고 결정을 내리는 능력임.
- AI(인공지능)이란, 인간의 지능이 필요한 작업을 기계도 할 수 있게 만드는 것을 목표로 하는 과학임.
- 기계가 특정 인지 작업에서 인간 수준의 성과를 낼 수 있으면 지능이 있다고 생각할 수 있음. 지능적인 기계를 만들기 위해서는 특정 문제 영역에서 전문가 지식을 획득하고 구성하며 사용할 수 있어야 함.
- 보다 지능적인 기계를 위해 약한 방법에서 특정 분야의 지식 집중적인 방법으로 '패러다임 전환'이 이루어졌음. 이는 한정된 문제 영역에서 인간 전문가 수준의 성능을 보이는 '전문가 시스템' 을 개발하기도 함.
- 전문가 시스템의 특징
  - 특정 규칙의 형태로 인간의 지식과 경험을 사용하고 지식이 분리되어 있음.
  - 자신의 추론 절차를 설명할 수 있음.
  - 전문가 시스템은 학습할 수 없고, 경험을 통해 자신을 개선시킬 수도 없음. 전문가 시스템은 개별적으로 구축되고, 개발하는 데 많은 노력이 필요함.

03_요약

❖ 지식기반 지능형 시스템 개론 요약

- 지능형 기계를 만들 때, 즉 지식 공학에서 겪는 주된 어려움 중 하나는 전문가에게 지식을 얻어내는 작업인 '지식 획득 병목현상'임.

- 생물학적 신경망에서 영감을 얻은 인공 신경망(ANN)은 과거의 경험에서 학습하고, 자동적으로 규칙을 생성해서 지식을 습득하여, 이를 확인하고 수정하는 과정을 피할 수 있음.

- 퍼지 논리(퍼지 집합론)은 단어로 계산하는 수단을 제공.
  - 단어의 의미, 인간의 추론, 결정을 알 수 있는 퍼지 값을 사용하는데 집중하고 전통적인 전문가 시스템의 계산적 부담을 해결하는 방법을 제공.

03_요약 – AI와 지식 공학의 주요 사건

[표 1-1] AI와 지식 공학의 역사에서 주목할 만한 주요 사건

기간	주요 사건
AI의 탄생 (1943년~1956년)	McCulloch and Pitts, *A Logical Calculus of the Ideas Immanent in Nervous Activity*, 1943 Turing, *Computing Machinery and Intelligence*, 1950 The Electronic Numerical Integrator and Calculator project (von Neumann) Shannon, Programming a Computer for Playing Chess, 1950 The Dartmouth College summer workshop on machine intelligence, artificial neural nets and automata theory, 1956
AI의 융성 (1956년~1960년대 후반)	*LISP* (McCarthy) The General Problem Solver (GPR) project (Newell and Simon) Newell and Simon, *Human Problem Solving*, 1972 Minsky, *A Framework for Representing Knowledge*, 1975
AI에 관한 환멸 (1960년대 후반~1970년대 초반)	Cook, The Complexity of Theorem Proving Procedures, 1971 Karp, Reducibility Among Combinatorial Problems, 1972 The Lighthill Report, 1971
전문가 시스템의 발명 (1970년대 초반~1980년대 중반)	DENDRAL (Feigenbaum, Buchanan and Lederberg, Stanford University) MYCIN (Feigenbaum and Shortliffe, Stanford University) PROSPECTOR (Stanford Research Institute) PROLOG-a Logic Programming Language (Colmerauer, Roussel and Kowalski, France) EMYCIN (Stanford University) Waterman, *A Guide to Expert Systems*, 1986

03_요약 – AI와 지식 공학의 주요 사건

기간	주요 사건
인공 신경망의 재탄생 (1965년~현재)	Hopfield, *Neural Networks and Physical Systems with Emergent Collective Computational Abilities*, 1982 Kohonen, *Self-Organized Formation of Topologically Correct Feature Maps*, 1982 Rumelhart and McClelland, *Parallel Distributed Processing*, 1986 The First IEEE International Conference on Neural Networks, 1987 Haykin, *Neural Networks*, 1994 Neural Network, MATLAB Application Toolbox (The MathWork, Inc.)
진화 연산 (1970년대 초반~현재)	Rechenberg, *Evolutionsstrategien-Optimierung Technischer Systeme Nach Prinzipien der Biologischen Information*, 1973 Holland, *Adaptation in Natural and Artificial Systems*, 1975 Koza, *Genetic Programming: On the Programming of the Computers by Means of Natural Selection*, 1992 Schwefel, *Evolution and Optimum Seeking*, 1995 Fogel, *Evolutionary Computation-Towards a New Philosophy of Machine Intelligence*, 1995
단어 계산 (1980년대 후반~현재)	Zadeh, *Fuzzy Sets*, 1965 Zadeh, *Fuzzy Algorithms*, 1969 Mamdani, *Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis*, 1977 Sugeno, *Fuzzy Theory*, 1983 Japanese 'fuzzy' consumer products (dishwashers, washing machines, air conditioners, television sets, copiers) Sendai Subway System (Hitachi, Japan), 1986 Negoita, *Expert Systems and Fuzzy Systems*, 1985

03_요약 – AI와 지식 공학의 주요 사건

기간	주요 사건
단어 계산 (1980년대 후반~현재)	The First IEEE International Conference on Fuzzy Systems, 1992 Kosko, *Neural Networks and Fuzzy Systems*, 1992 Kosko, *Fuzzy Thinking*, 1993 Yager and Zadeh, Fuzzy Sets, *Neural Networks and Soft Computing*, 1994 Cox, *The Fuzzy Systems Handbook*, 1994 Kosko, Fuzzy Engineering, 1996 Zadeh, *Computing with Words-A Paradigm Shift*, 1996 Fuzzy Logic, MATLAB Application Toolbox (The MathWork, Inc.)

Thank You !