

# 1 | 문제 해결

## 1 인공지능과 문제 해결

### 문제

- ▶ 우리가 살아가면서 만나는 다양한 문제(problem)들을 해결
- ▶ 인공지능이 발전하더라도 인간이 해결해야 할 일도 많음
  - 인공지능이 하나의 하드웨어가 되고 인간은 소프트웨어로써 인공지능을 활용
  - 인간의 생각과 사고가 없다면 인공지능도 큰 역할을 수행하지 못함
- ▶ 인공지능 시대에서 문제 해결에 관한 논제는 매우 중요
- ▶ 어떤 방법으로 문제를 해결할지에 대한 깊은 사고력 필요

## 1 인공지능과 문제 해결

### 문제 해결에 도움이 되는 사고

- ▶ 기본 개념과 원리를 생각하며 문제에 적용 (basics)
- ▶ 비슷한 유형의 문제는 같은 방법으로 적용 (similarity)
- ▶ 어렵게 보이는 문제를 단순화시킬 수 있는지를 점검 (abstraction)
- ▶ 복잡한 문제는 여러 단계로 나누어 문제 해결 (decomposition)
- ▶ 다양한 관점으로 문제의 핵심에 접근 (multiformity)
- ▶ 자유로운 생각으로 문제 해결의 실마리를 끌어냄 (creativity)

## 1 인공지능과 문제 해결

### 인공지능 시대의 문제 해결 전략

- ▶ 컴퓨터를 이용하여 문제 해결이 가능한 방안 마련
- ▶ 블록 다이어그램을 그려 문제를 단계별로 분석
- ▶ 규칙을 찾아 규칙기반 인공지능에 적용 여부 고려
- ▶ 신경망이나 딥러닝의 인식 기능 활용
- ▶ 데이터 사이언스나 빅데이터를 적용할 수 있는지 검토
- ▶ 인간의 사고와 컴퓨터 능력을 통합한 컴퓨팅 사고 적용

## 2 문제 해결의 핵심인 사고의 힘

### 사고력

- ▶ 인간 사고 능력의 결과물은 지식
- ▶ 지식을 오랜 기간 문자를 통해 후대로 전달
- ▶ 연산의 필요로 컴퓨터가 발명
  - 인간처럼 사고할 수 있는 인공지능에 도전
- ▶ 인공지능의 발전에도 불구하고 인간의 사고력은 중요한 역할

## 3 인공지능 시대의 컴퓨팅 사고

### 컴퓨팅 사고

- ▶ 인공지능 시대에는 새로운 사고 방법 필요
  - 단순한 방식의 문제 해결과 맹목적인 코딩이 불필요
- ▶ 컴퓨팅 사고(computational thinking) 대두
- ▶ 컴퓨터를 활용하여 문제 해결을 위해 사고하는 방법
- ▶ 기존의 인간 사고와는 달리 컴퓨터를 활용하여 문제를 해결하는 과정에서 여러 가지 특성을 포함하는 문제 해결 과정

## 3 인공지능 시대의 컴퓨팅 사고

### 컴퓨팅 사고

#### ▶ 지넷 윙(Wing) 교수 주장

컴퓨팅 사고는 누구에게나 일반적으로 적용되는 사고 방식과 기술의 집합인데, 배우고 익혀서 사용할 가치가 충분

## 3 인공지능 시대의 컴퓨팅 사고

### 컴퓨팅 사고

- ▶ 인간의 사고 능력과 컴퓨터의 능력을 통합한 사고
- ▶ 복합적 사고를 통해 창의적으로 문제를 해결하는 능력
- ▶ 복잡한 문제의 해결에 상당히 효율적인 결과 보여줌
- ▶ 복잡한 것을 자신감 있게 다루는 능력 향상
- ▶ 어려운 문제들을 지속적으로 다루는 능력 향상
- ▶ 다양한 해답이 있을 수 있는 문제를 다루는 능력 향상

## 3 인공지능 시대의 컴퓨팅 사고

### 컴퓨팅 사고에서 고려해야 할 주요 사항들

- ▶ 문제 해결의 결과 정확한 답을 얻을 수 있는가?
- ▶ 가장 효율적인 문제 해결 방법인가?
- ▶ 가장 빠르고 논리적인 문제 해결 방법인가?
- ▶ 다른 문제들을 해결하는데도 쓰일 수 있는가?

## 3 인공지능 시대의 컴퓨팅 사고

### 컴퓨팅 사고의 발상

- ▶ 컴퓨팅 사고는 컴퓨터 능력 + 깊은 사고력의 융합
- ▶ 문제를 효율적으로 해결하기 위한 방법
- ▶ 문제 분석, 분해, 패턴 찾아내기, 추상화 단계를 거쳐 알고리즘으로 연결
- ▶ 프로그래밍을 통해 컴퓨터를 작동시켜 자동화 구현

[컴퓨팅 사고의 발상]

컴퓨팅 능력  
(Power of computing)



깊은 사고력  
(Critical thinking)



컴퓨팅 사고  
(Computational thinking)

## 4 문제 해결을 위한 컴퓨팅 사고

### 문제 해결을 위한 컴퓨팅 사고의 4대 요소

- ▶ 분해, 패턴인식, 추상화, 그리고 알고리즘 단계
- ▶ 이 4가지 요소들을 정확하게 활용하는 기술

#### [컴퓨팅 사고의 4대 요소]

분해  
(Decomposition)

패턴인식  
(Pattern Recognition)

추상화  
(Abstraction)

알고리즘  
(Algorithm)

## 4 문제 해결을 위한 컴퓨팅 사고

### 컴퓨팅 사고의 4단계

#### ▶ 분해

- 어려운 문제를 작은 단위의 쉬운 문제로 나누어 해결할 수 있는 사고
- 복잡한 난이도의 감소를 유도

#### ▶ 패턴인식

- 규칙성과 패턴들을 발견, 유사성 찾아내기
- 문제를 효율적으로 해결하기 위함

## 4 문제 해결을 위한 컴퓨팅 사고

### 컴퓨팅 사고의 4단계

#### ▶ 추상화

- 복잡한 문제의 핵심을 파악할 수 있는 사고
- 문제 내 필요한 부분과 필요하지 않은 부분을 분리하여 집중하는 것

#### ▶ 알고리즘

- 문제들을 해결할 수 있는 일련의 논리적인 지시
- 문제에 대한 단계적인 해결책이나 설명, 지시 사항들을 설계

## 5

### 컴퓨팅 사고의 장점

- ▶ 문제 해결 능력 배양
- ▶ 일반적인 지식은 필요한 분야에만 사용이 한정되는 경우가 많음
- ▶ 컴퓨팅 사고에서는 사고의 템플릿(template)을 여러 분야에 적용 가능
  - 컴퓨팅 사고의 템플릿을 익히면 여러 분야에 활용 가능
  - 컴퓨터 분야의 컨텐츠 개발 및 다양한 공학 분야나 문학 등 인문 분야에도 문제 해결 과정에 큰 도움
- ▶ 지식의 효율적 활용 및 인공지능 기술과의 융합 기대

## 5 컴퓨팅 사고의 장점

### 컴퓨팅 사고를 통한 문제 해결의 예

‘서울버스’ 앱

- ▶ 무작정 버스를 기다리는 불편함 해소
- ▶ 컴퓨팅 사고의 방법을 잘 적용
  - 버스의 출발 시각, 주행 시간, 대기 시간 등의 자료를 수집하고 분석
  - 복잡함을 단순하게 만들고 컴퓨팅 사고의 과정을 거쳐 적용
- ▶ 버스 이용객에게 많은 사랑을 받고 있음

## 5 컴퓨팅 사고의 장점

### 컴퓨팅 사고를 통한 문제 해결의 예

#### 심야 버스 노선 신설

- ▶ 사람들이 심야에 택시를 많이 호출하는 지역에 심야 노선 버스를 신설
- ▶ 원인 파악에서 해결 방안으로 컴퓨팅 사고를 접목
  - 심야 시간 택시 호출을 위한 전화 통화량 수집
  - 심야 택시의 이동 경로 수집

## 2 | 현대의 코딩

### 1 코딩을 통한 인공지능 접근

- ▶ 컴퓨팅 사고의 영향으로 직접 프로그래밍을 하는 코딩이 주목
- ▶ 코딩으로 만들어지는 앱(App) 또는 어플리케이션에 대한 기초적인 지식 필요
- ▶ 코딩
  - 컴퓨터 프로그램을 수행하는 절차를 적어둔 명령어들인 코드(code)를 작성하는 행위
  - 실행 가능한 프로그램을 작성하는 일
- ▶ 알고리즘
  - 코딩을 위해 명령어들을 순서대로 나열하는 작업
  - 논리적이고 효율적이어야 함

### 1 코딩을 통한 인공지능 접근

#### 코딩의 목표

- ▶ 주어진 문제를 제대로 해결하는 일 (문제 해결)
- ▶ 코딩하기 전에 문제 해결을 위한 방법부터 먼저 구상 (알고리즘)
- ▶ 코딩을 마치고 컴퓨터를 작동시켜 문제 해결 (자동화)

[문제 해결과 코딩]



### 1 코딩을 통한 인공지능 접근

#### 인공지능 문제에 대한 코딩

- ▶ 현재 인공지능에 주 관심사는 기계학습과 딥러닝, 인식 등
- ▶ 기계 학습을 위해 필요한 빅데이터를 처리하기 위해 자동화된 시스템 필요
- ▶ 학습 목적에 맞는 자동화된 시스템을 개발하기 위해 코딩이 접목
- ▶ 형식적 언어가 아닌 즉시 응답을 받을 수 있는 스크립트 형 언어들이 주목
  - 파이선이나 R과 같은 언어들은 쉬운 문법과 다양한 기능 활용 가능
  - 데이터 수집부터 분석, 처리, 표현까지 다양한 활용이 가능
  - 누구나 쉽게 배우고 목적에 맞는 학습 알고리즘의 개발이 가능
  - 코딩에 필요한 기본적인 개념과 활용 능력을 학습하여 다양하게 활용 가능

### 2 코딩의 중요성

#### ▶ 빌 게이츠(Bill Gates)

13살 때 처음으로 코딩하는 방법을 배웠고, 이를 바탕으로 마이크로소프트를 세웠다.

#### ▶ 미국 오바마(Obama) 전 대통령

코딩 기술을 배우는 것이 여러분의 미래는 물론 조국의 미래에도 매우 중요하다.

#### ▶ 스티브 잡스(Steve Jobs)

이 나라 모든 사람들은 컴퓨터 프로그래밍 즉 코딩을 배워야 한다. 코딩은 생각하는 방법을 가르쳐주기 때문이다.

### 2 코딩의 중요성

- ▶ 디지털 시대에 필요한 컴퓨팅적 사고력을 키우는 교육 도구
- ▶ 현재는 4차 산업 시대이며 4차 산업 시대의 핵심 요소는 “융합”
- ▶ 융합 시대에 산업과 산업을 융합 시킬 수 있는 연결 고리가 코딩이 될 수 있음

### 2 코딩의 중요성

#### 4차 산업 시대의 인공지능과 코딩

- ▶ 소프트웨어 전공자들에게만 필요한 코딩이 아닌 다른 분야의 전공자들에게도 필요
- ▶ 개발자 코딩 교육과 사용자 코딩 교육은 구분되어야 함
- ▶ 인공지능을 이용한 소프트웨어 개발도 가능하게 된 시대에 모든 학생들이 개발을 위한 코딩 교육을 받을 필요는 없음
- ▶ 단지 컴퓨팅 사고를 통한 문제 해결 능력을 키워 4차 산업 시대의 인공지능을 활용하는 방식으로 변화가 필요

### 3 코딩의 주요 목적

- ▶ 인공지능 시대에는 코딩이 기초적이고 필수적인 요소
- ▶ 코딩 과정을 통해 논리적 사고력 향상
- ▶ 문제 해결을 위한 생각하는 힘을 기를 수 있는 장점
- ▶ 코딩을 함으로써 창의적 발상과 지구력 향상
- ▶ 컴퓨팅 사고로 미래 직업에 대한 효율적인 대응 전략

### 4 코딩의 종류

#### 스크래치(scratch)

- ▶ 2007년 미국 MIT에서 개발된 무료 소프트웨어
- ▶ 프로그래밍 초보자용 시각적 교육용 프로그래밍 언어
- ▶ 여러 가지 모양과 색깔의 코드 블록들을 끌어다가 조합하는 블록형 프로그래밍
- ▶ 캐릭터가 춤을 추게 하거나 스토리를 만들 수 있음
- ▶ 게임, 애니메이션, 음악, 미술, 대화형 스토리텔링 등 구현
- ▶ 누구나 컴퓨터 프로그래밍의 개념을 쉽게 배울 수 있음

### 4 코딩의 종류

#### 아두이노(Arduino)

- ▶ 마이크로 보드와 같은 하드웨어를 조절하는 코딩 도구의 예
- ▶ 아두이노는 다양한 제작물을 만드는 개방형 프로그램

#### 아두이노 활용 예

- 키보드나 마우스로 조종하는 움직이는 자동차 로봇 구현
- IoT 센서를 활용한 기기 제어
- 블루투스 등을 활용한 무선 통신 기술 접목

### 4 코딩의 종류

#### Python

- ▶ 1991년 귀도 로섬이 발표한 프로그래밍 언어
- ▶ 누구나 [www.python.org](http://www.python.org)에서 무료로 다운받아 사용 가능한 오픈 소스
- ▶ C, Java, 그리고 LISP의 특징들을 모두 가지고 있음
- ▶ 인공지능의 확산으로 인기 있는 프로그래밍 언어
- ▶ 최근 사용자가 꾸준히 늘고 있는 추세

### 4 코딩의 종류

#### Python

##### ▶ Python의 주요 특징 요약

- 인터프리터 언어로서 실행 결과를 즉석에서 확인할 수 있는 스크립트 언어
- 다른 언어에 비해 문법이 비교적 간단
- ‘들여쓰기’로 블록을 구분하는 독특한 문법을 채용
- 포인터 개념이 없음
- 신경망과 딥러닝 코딩에 많이 쓰임
- 텐서플로가 Python 응용 프로그래밍 인터페이스(API) 제공

### 3 | 인공지능 코딩의 경험

## 1 인공지능 코딩의 사용 예

규칙 기반 문제 해결을 위한 코딩	<ul style="list-style-type: none"><li>■ 데이터베이스에 규칙을 설정</li><li>■ 문제가 주어지면 문제에서 규칙을 적용할 키워드를 추출</li><li>■ 키워드에 해당하는 규칙을 데이터베이스에서 검색하여 결과 도출</li></ul>
확률 기반 문제 해결을 위한 코딩	<ul style="list-style-type: none"><li>■ 사전에 발생하는 사건들에 대한 확률을 계산</li><li>■ 문제가 주어지면 문제에서 발생하는 사건들을 확률과 통계를 산출</li><li>■ 문제 해결에 필요한 확률 값을 계산하여 결론 도출</li></ul>
분류, 회귀와 같은 기계 학습 문제 해결을 위한 코딩	<ul style="list-style-type: none"><li>■ 주어진 문제의 도메인에서 목적에 따른 타겟 값을 구분하여 데이터베이스에 저장</li><li>■ 문제가 주어지면 해당하는 타겟 값을 계산 식에 의한 산출</li></ul>

## 1 인공지능 코딩의 사용 예

### 규칙 기반 문제 해결을 위한 코딩

#### ▶ LISP의 예

- 주어진 도형이 원(circle)인지 구체(sphere) 인지에 따른 면적 계산 예

```
? (setq thing sphere r 1)
1
? (cond ((equal thing' circle (* 3.14159 r r))
          ((equal thing' sphere (* 4 3.14159 r r)))
12.56637
```

## 1 인공지능 코딩의 사용 예

### 규칙 기반 문제 해결을 위한 코딩

#### ▶ Prolog의 예

- 기차 출발지에서 목적지까지의 이동 시간 계산

기차 시각표	규칙
시각표(출발, 도착, 출발시간, 도착시간) 시각표(서울, 대전, 800, 930) 시각표(서울, 대전, 900, 1030) 시각표(대전, 광주, 940, 1240) 시각표(대전, 광주, 1040, 1340)	계획 (출발, 도착, 출발시간, 도착시간) :- 시각표(출발, 도착, 출발시간, 도착시간) 계획 (출발, 도착, 출발시간, 도착시간) :- 시각표 (출발, 환승, 출발시간, 환승 도착시간), 시각표 (환승, 도착, 환승 출발시간, 도착시간), > (환승 출발시간, 환승 도착시간)

[질문] ? - 계획 (서울, 광주, 출발시간, 도착 시간)  
< 도착 시간, 1300>

[결과] 출발시간 = 800  
도착시간 = 1240

## 1 인공지능 코딩의 사용 예

### 확률 기반 문제 해결을 위한 코딩

#### ▶ 나이브 베이즈 분류기 예

##### ■ 스팸 메일 분류기

-	단어	분류
1	me free lottery	스팸
2	free get free you	스팸
3	you free scholarship	정상
4	free to contact me	정상
5	you won award	정상
6	you ticket lottery	스팸

### 3 | 인공지능 코딩의 경험

## 1 인공지능 코딩의 사용 예

### 확률 기반 문제 해결을 위한 코딩

#### ▶ 나이브 베이즈 분류기 예

- 스팸 메일 분류기

정상메일일 확률 < 스팸 메일일 확률

[공식]

$$P(\text{정상}|\text{텍스트}) = P(w_1|\text{정상}) * P(w_2|\text{정상}) * P(w_3|\text{정상}) * P(\text{정상})$$

$$P(\text{스팸}|\text{텍스트}) = P(w_1|\text{스팸}) * P(w_2|\text{스팸}) * P(w_3|\text{스팸}) * P(\text{스팸})$$

[문제]

도착한 메일에 free, you, lottery 라는 단어가 있는 경우 스팸메일로 분류될 수 있는 확률은 얼마나 되는가?

[계산]

$$P(\text{정상}|you, \text{free}, \text{lottery}) = P(you|\text{메일}) * P(\text{free}|\text{정상}) * P(\text{lottery}|\text{정상}) * P(\text{정상})$$

$$P(\text{스팸}|you, \text{free}, \text{lottery}) = P(you|\text{스팸}) * P(\text{free}|\text{스팸}) * P(\text{lottery}|\text{스팸}) * P(\text{스팸})$$

## 1 인공지능 코딩의 사용 예

### 확률 기반 문제 해결을 위한 코딩

- ▶ GMM(가우시안 혼합 모델)
  - 문서의 주제에 따른 분류
  - 퍼지 영상 분할(Fuzzy image segmentation)

### 3 | 인공지능 코딩의 경험

## 1 인공지능 코딩의 사용 예

### 분류, 회귀와 같은 기계 학습 문제 해결을 위한 코딩

#### ▶ 나이와 몸무게의 추정 예

- 16명의 나이와 키 데이터를 수집

나이	15.43	23.01	5	12.56	8.67	7.31	9.66	13.64	14.92	18.47	15.48	22.13	10.11	26.95	5.68	21.76
키	170.91	160.68	129	159.7	155.46	140.56	153.65	159.43	164.7	169.65	160.71	173.29	159.31	171.52	138.96	165.87

[문제] 나이가 17살인 학생의 키는 얼마라고 예측하는가?

산출된 값: 161.6

## 1 인공지능 코딩의 사용 예

### 분류, 회귀와 같은 기계 학습 문제 해결을 위한 코딩

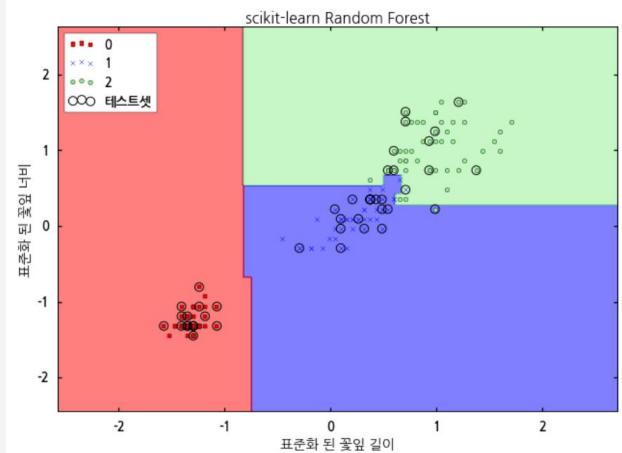
#### ▶ 봇꽃 분류기

```
X, y = make_moons(n_samples=100, noise=0.25, random_state=3)
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, random_state=42)
forest = RandomForestClassifier(n_estimators=5, random_state=2)
forest.fit(X_train, y_train)

fig, axes = plt.subplots(2, 3, figsize=(20, 10))
for i, (ax, tree) in enumerate(zip(axes.ravel(), forest.estimators_)):
    print(i)
    print(ax)
    ax.set_title("tree {}".format(i))
    mglearn.plots.plot_tree_partition(X, y, tree, ax=ax)

mglearn.plots.plot_2d_separator(forest, X, fill=True, ax=axes[-1, -1], alpha=.4)
axes[-1, -1].set_title("random forest")
mglearn.discrete_scatter(X[:, 0], X[:, 1], y)
```

[파이썬 코드 예]



[랜덤포레스트 분류 결과]