

인공지능의 이해

Understanding Artificial Intelligence

유준수 (Joon Soo Yoo, Ph.D.)

Fall 2026

Defense & Advanced Security Lab (DAS Lab)

Department of Advanced Defense Technology and Industry

Jeonbuk National University

Introduction to Artificial Intelligence

From AI and Machine Learning
to Agents and Physical AI

Week 1

1. What is Artificial Intelligence?

Artificial Intelligence (AI)는 무엇일까?

우리는 이미 다양한 AI 시스템을 사용하고 있다.

- ▶ ChatGPT
- ▶ Spam Filter
- ▶ Recommendation System
- ▶ Face Recognition
- ▶ Autonomous Vehicle
- ▶ Robot
- ▶ Game AI

이 시스템들은 서로 매우 다르게 보인다.

하지만 공통적으로 컴퓨터가

Information을 이용하여 결과를 만들거나 행동을 결정한다.

핵심

AI는 컴퓨터가 정보를 이용하여 Prediction, Decision, 또는 Action을 수행하게 만드는 기술이다.

1-1. Is This AI?

다음 시스템들은 AI일까?

System	AI?
Calculator	?
Spam Filter	?
Chess Program	?
ChatGPT	?
Robot Vacuum	?
Autonomous Vehicle	?

AI의 경계는 생각보다 명확하지 않다.

시대에 따라 “Intelligent”하다고 생각하는 기술의 기준도 계속 변화해왔다.

핵심

AI는 하나의 특정 알고리즘을 의미하는 것이 아니라, 지능적인 문제 해결을 위한 넓은 기술 영역이다.

1-2. AI Is Broader Than Machine Learning

AI와 Machine Learning은 같은 의미가 아니다.

Artificial Intelligence



Search

Planning

Rule-based Systems

Machine Learning

초기의 AI는 반드시 데이터를 학습하지 않았다.

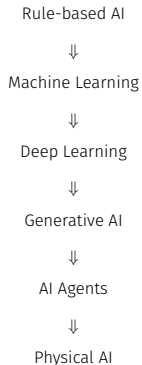
사람이 규칙을 만들거나, 가능한 답을 탐색하여 문제를 해결하기도 했다.

핵심

Machine Learning은 Artificial Intelligence를 구현하는 여러 방법 중 하나이다.

2. How has AI Expanded

AI 기술은 계속 변화해왔다.



각 단계가 이전 기술을 완전히 대체하는 것은 아니다.

새로운 방법들이 추가되면서 AI가 해결할 수 있는 문제의 범위가 넓어지고 있다.

2-1. From Rules to Learning

전통적인 프로그램에서는 사람이 규칙을 작성한다.

Data + Rules



Program



Output

예를 들어,

Source Code

```
if temperature > 38:  
    print("High Temperature")
```

하지만 모든 문제의 규칙을 사람이 직접 작성하기는 어렵다.

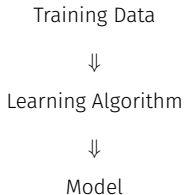
사진을 보고 고양이인지 판단하는 규칙을 수천 개의 `if` 문으로 작성할 수 있을까?

핵심

복잡한 문제에서는 사람이 모든 규칙을 직접 정의하기 어렵다. 이러한 한계가 Learning-based Approach의 중요성을 높였다.

2-2. Machine Learning

Machine Learning에서는 컴퓨터가 Data로부터 Pattern을 학습한다.



New Data → Model → Prediction

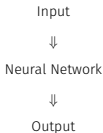
대표적인 문제는 다음과 같다.

- ▶ Regression: 숫자를 예측한다.
- ▶ Classification: Category를 예측한다.
- ▶ Clustering: 비슷한 Data를 그룹으로 나눈다.

2-3. Deep Learning

Deep Learning은 Machine Learning의 한 분야이다.

Deep Learning에서는 여러 Layer로 구성된 Neural Network를 사용한다.



예를 들어 이미지가 입력되면,

Image → Neural Network → Cat: 0.92

Deep Learning은 특히 다음 분야의 발전에 큰 영향을 주었다.

- ▶ Computer Vision
- ▶ Speech Recognition
- ▶ Natural Language Processing

핵심

Deep Learning은 Neural Network를 이용하여 복잡한 Pattern을 Data로부터 학습한다.

2-4. Generative AI

기존의 많은 AI는 주어진 입력을 분석하고 예측하는 데 집중했다.

Generative AI는 새로운 Content를 생성할 수 있다.

Input	Generated Output	Example
Text	Text	Question → Answer
Text	Image	Prompt → Generated Image
Text	Code	Request → Python Code
Audio	Text	Lecture Audio → Summary
Image + Text	Text	Image + Question → Description

대표적인 예가 Large Language Model (LLM)이다.

Prompt → LLM → Generated Response

핵심

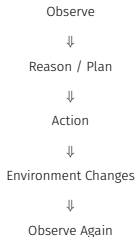
Generative AI는 학습한 Pattern을 이용하여 새로운 Text, Image, Code 등의 Content를 생성한다.

2-5. From AI Models to AI Agents

AI Model은 일반적으로 입력을 받아 출력을 만든다.

Input → AI Model → Output

AI Agent는 Environment와 반복적으로 상호작용한다.



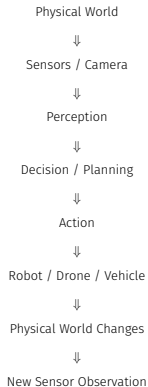
이 과정은 목표를 달성할 때까지 반복될 수 있다.

핵심

AI Agent는 단순히 한 번의 출력을 만드는 것이 아니라, Environment를 관찰하고 행동하며 다시 결과를 관찰하는 Closed-loop 구조로 동작할 수 있다.

2-6. From AI Agents to Physical AI

Physical AI에서는 AI의 Action이 실제 Physical World를 변화시킨다.



핵심

Physical AI는 Perception, Decision, Action이 Physical World를 통해 다시 Sensor 입력으로 연결되는 Closed-loop Intelligence이다.

3. The AI Landscape

이번 학기에는 AI의 전체적인 흐름을 따라간다.

Artificial Intelligence



Machine Learning



Deep Learning



Generative AI / LLM



AI Agents



Physical AI

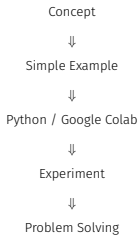
각 기술을 깊게 구현하는 것보다,

- ▶ 무엇을 해결하는 기술인지
- ▶ 어떤 원리로 동작하는지
- ▶ 어떻게 평가하는지
- ▶ 실제 문제에 어떻게 사용할 수 있는지

를 이해하는 것이 이번 수업의 목표이다.

3-1. How We Will Learn AI

이번 수업에서는 이론만 배우지 않는다.



실습에서는 완성된 코드를 처음부터 작성하기보다,

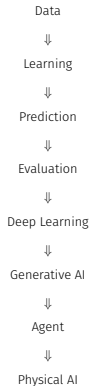
- ▶ 코드를 실행하고
- ▶ 일부 값을 변경하고
- ▶ 결과를 관찰하고
- ▶ 결과의 의미를 해석하고
- ▶ 간단한 문제를 해결한다.

핵심

Programming 자체보다 시가 어떻게 동작하고 문제를 어떻게 해결하는지 이해하는 것이 중요하다.

3-2. Our Journey

이번 학기의 전체 흐름은 다음과 같다.



핵심 문장

AI가 Data로부터 어떻게 학습하고, 어떻게 판단하며, 궁극적으로 Physical World에서 어떻게 행동하는지를 이해한다.

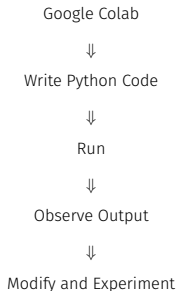
Hands-on Practice

Google Colab and Python Basics

4. Practice: Google Colab and Python Basics

이번 실습에서는 AI 실습에 필요한 가장 기본적인 Python 사용법을 익힌다.

실습은 Google Colab에서 진행한다.



실습 목표

Python 문법을 깊게 배우는 것이 아니라, AI 실습에 필요한 최소한의 코드를 읽고, 실행하고, 수정하는 방법을 익힌다.

4-1. What is Google Colab?

Google Colab은 Web Browser에서 Python 코드를 실행할 수 있는 환경이다.

별도의 Python 설치 없이 사용할 수 있다.

- ▶ Web Browser에서 실행
- ▶ Python 사용 가능
- ▶ Notebook 형태로 코드 작성
- ▶ Data Analysis 가능
- ▶ Machine Learning Library 사용 가능
- ▶ GPU 사용 가능

Browser → Google Colab → Python Runtime

핵심

이번 수업에서는 복잡한 개발 환경 설정 대신, Google Colab을 이용하여 AI 실습을 진행한다.

4-2. Your First Python Code

Colab Notebook의 Code Cell에 다음 코드를 입력한다.

Source Code

```
print("Hello, AI!")
```

실행 버튼을 누르면 다음 결과가 나타난다.

Output

```
Hello, AI!
```

`print()`는 화면에 값을 출력하는 Python 함수이다.

핵심

Python 코드는 Code Cell에 작성하고, 실행하면 바로 아래에서 결과를 확인할 수 있다.

4-3. Variables: 변수

Variable은 값을 저장하기 위한 이름이다.

Source Code

```
name = "Alice"  
age = 20  
score = 85.5  
  
print(name)  
print(age)  
print(score)
```

각 Variable에는 서로 다른 종류의 값을 저장할 수 있다.

Variable	Value	Type
name	"Alice"	String
age	20	Integer
score	85.5	Float

핵심

Variable을 사용하면 Data에 이름을 붙여 저장하고 다시 사용할 수 있다.

4-4. Numbers and Basic Operations

Python에서는 기본적인 수학 연산을 사용할 수 있다.

Source Code

```
a = 10  
b = 3  
  
print(a + b)  
print(a - b)  
print(a * b)  
print(a / b)
```

실행 결과는 다음과 같다.

Output

```
13  
7  
30  
3.3333333333333335
```

Operator	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	Division

4-5. Boolean and Comparison

Python은 두 값을 비교할 수 있다.

Source Code

```
score = 85  
  
print(score >= 80)  
print(score < 80)  
print(score == 85)
```

실행 결과는 다음과 같다.

Output

```
True  
False  
True
```

*True*와 *False*는 Boolean 값이라고 한다.

핵심

Comparison은 어떤 조건이 맞는지 확인하며, 그 결과는 *True* 또는 *False*가 된다.

4-6. Lists

List는 여러 개의 값을 하나로 묶어 저장한다.

Source Code

```
scores = [70, 80, 90, 95]

print(scores)
print(scores[0])
print(scores[2])
```

실행 결과는 다음과 같다.

Output

```
[70, 80, 90, 95]

70

90
```

Python의 Index는 0부터 시작한다.

Index	0	1	2	3
Value	70	80	90	95

핵심

AI에서는 많은 Data를 다루기 때문에, 여러 값을 저장하는 구조가 중요하다.

4-7. Simple Calculation with Lists

List의 값을 이용하여 간단한 계산을 할 수 있다.

Source Code

```
scores = [70, 80, 90, 95]

total = sum(scores)
count = len(scores)
average = total / count

print(total)
print(count)
print(average)
```

실행 결과는 다음과 같다.

Output

```
335
4
83.75
```

- ▶ `sum()`: List의 값을 모두 더한다.
- ▶ `len()`: List에 몇 개의 값이 있는지 계산한다.

핵심

Data를 저장하는 것뿐 아니라, Data를 계산하고 요약하는 것도 AI 실습의 기본 과정이다.

4-8. Conditional Statements

`if` 문을 사용하면 조건에 따라 다른 동작을 수행할 수 있다.

Source Code

```
score = 85

if score >= 80:
    print("Pass")
else:
    print("Fail")
```

실행 결과

Output

```
Pass
```

Python에서는 들여쓰기(Indentation)가 중요하다.

핵심

Conditional Statement는 조건에 따라 Program의 행동을 결정한다.

4-9. Repetition with for

`for` 문은 여러 Data에 대해 같은 작업을 반복할 때 사용한다.

Source Code

```
scores = [70, 80, 90, 95]

for score in scores:
    print(score)
```

실행 결과

Output

```
70
80
90
95
```

각 List의 값을 하나씩 꺼내어 같은 코드를 반복해서 실행한다.

핵심

Loop를 사용하면 많은 Data에 대해 같은 작업을 반복할 수 있다.

4-10. Combining for and if

`for`와 `if`를 함께 사용할 수 있다.

Source Code

```
scores = [70, 80, 90, 95]

for score in scores:
    if score >= 90:
        print(score)
```

실행 결과

Output

```
90
95
```

전체 Data를 하나씩 확인하면서 조건을 만족하는 Data만 선택한 것이다.

핵심

Loop와 Condition을 결합하면 Data를 반복적으로 확인하고 필요한 결과만 선택할 수 있다.

4-11. Practice: AI Predictions

어떤 AI Model이 다음과 같은 Prediction을 만들었다고 하자.

Source Code

```
predictions = [0, 1, 1, 0, 1]
answers      = [0, 1, 0, 0, 1]
```

여기서

- ▶ *predictions*: AI Model이 예측한 값
- ▶ *answers*: 실제 정답

두 List를 비교해보자.

Index	Prediction	Answer
0	0	0
1	1	1
2	1	0
3	0	0
4	1	1

4-12. Is the Prediction Correct?

AI Model이 하나의 Prediction을 만들었다고 생각해보자.

Source Code

```
prediction = 1
answer = 1

print(prediction)
print(answer)
```

Prediction과 실제 Answer가 같으면 올바른 Prediction이다.

Source Code

```
if prediction == answer:
    print("Correct")
else:
    print("Wrong")
```

실행 결과

Output

```
Correct
```

핵심

AI의 Prediction은 실제 Answer와 비교하여 맞았는지 틀렸는지 확인할 수 있다.

4-13. Several AI Predictions

이번에는 AI가 여러 개의 Prediction을 만들었다고 하자.

Source Code

```
predictions = [0, 1, 1, 0, 1]
answers      = [0, 1, 0, 0, 1]

print(predictions)
print(answers)
```

직접 Prediction과 Answer를 비교해보자.

Prediction	Answer	Result
0	0	Correct
1	1	Correct
1	0	Wrong
0	0	Correct
1	1	Correct

총 5개의 Prediction 중 4개가 맞았다.

핵심

AI Model의 결과도 실제 정답과 비교하여 얼마나 잘 동작하는지 평가할 수 있다.

4-14. A First Look at Accuracy

앞에서 AI는 총 5개 중 4개를 맞혔다.

Python으로 간단히 계산해보자.

Source Code

```
correct = 4
total = 5

accuracy = correct / total

print(accuracy)
```

실행 결과

Output

```
0.8
```

즉,

$$Accuracy = \frac{4}{5} = 0.8 = 80\%$$

핵심

Accuracy는 전체 Prediction 중 얼마나 많은 Prediction을 맞혔는지를 나타낸다.

4-15. Mini Problem: Temperature

다음 Temperature Data가 있다.

Source Code

```
temperatures = [22, 35, 28, 40, 31]
```

먼저 모든 Temperature를 출력해보자.

Source Code

```
for temperature in temperatures:  
    print(temperature)
```

이제 다음 문제를 해결해보자.

1. 각 Temperature를 하나씩 확인한다.
2. 30도 이상이면 "Hot"을 출력한다.
3. 그렇지 않으면 "Normal"을 출력한다.

실습 목표

앞에서 배운 List, for, if를 하나의 간단한 문제에서 사용해본다.

4-16. Mini Problem: Example Solution

앞에서 배운 *for*와 *if*를 함께 사용한다.

Source Code

```
temperatures = [22, 35, 28, 40, 31]

for temperature in temperatures:

    if temperature >= 30:
        print(temperature, "Hot")

    else:
        print(temperature, "Normal")
```

실행 결과

Output

```
22 Normal
35 Hot
28 Normal
40 Hot
31 Hot
```

Program은 Temperature를 하나씩 확인하고, 정해진 Rule에 따라 결과를 출력한다.

핵심

List의 Data를 하나씩 확인하면서 Condition에 따라 서로 다른 결과를 만들 수 있다.

4-17. Try It Yourself: Change the Rule

이번에는 코드를 직접 조금 수정해보자.

현재 Rule은 다음과 같다.

Source Code

```
if temperature >= 30:  
    print("Hot")  
else:  
    print("Normal")
```

다음과 같이 바꾸어 실행해보자.

1. 기준을 30에서 35로 변경한다.
2. 기준을 25로 변경한다.
3. Temperature Data의 숫자를 직접 변경한다.

질문:

Rule을 바꾸면 Output은 어떻게 달라지는가?

실습 목표

처음부터 코드를 작성하기보다, 기존 코드를 수정하면서 Program의 동작을 이해한다.

4-18. Is This Machine Learning?

방금 만든 Temperature Program을 다시 생각해보자.

Source Code

```
if temperature >= 30:  
    print("Hot")  
else:  
    print("Normal")
```

여기서 가장 중요한 질문은 다음과 같다.

Who decided "30"?

정답은 사람이다.

Human defines the Rule



Temperature ≥ 30



Hot

핵심

이 Program은 Machine Learning이 아니다. 사람이 판단 Rule을 직접 정의한 Rule-based Program이다.

4-19. From Rules to Learning

오늘 실습에서는 사람이 직접 Rule을 만들었다.

Today: Rule-based Program

Data + Human-defined Rule



Program



Output

그렇다면 다음 질문을 생각해보자.

Can a computer learn the rule
from data?

Next: Machine Learning

Training Data



Learning



Model

Next Step

다음 단계에서는 사람이 Rule을 직접 정하지 않고, Computer가 Data로부터 Pattern을 학습하는 방법을 살펴본다.

Thank You

Questions?