# CoE202
# Fundamentals of Artificial intelligence
# <Big Data Analysis and Machine Learning>

## Introduction

Prof. Young-Gyu Yoon
School of EE, KAIST

# Instructor & TAs

- Instructor: Prof. Young-Gyu Yoon
  - Email: ygyoon@kaist.ac.kr

- Teaching assistants
  - Kim, Jeewon (김지원) one1won@kaist.ac.kr
  - Han, Seungjae (한승재) jay0118@kaist.ac.kr
  - Shin, Changyeop (신창엽) scey26@kaist.ac.kr
  - Eom, Minho (엄민호) djaalsgh159@kaist.ac.kr
  - Shul, Yusun (설유선) shulys@kaist.ac.kr
  - Yi, Wonjun (이원준) waganawa@kaist.ac.kr
  - Jeong, Namho (정남호) nhjeong@kaist.ac.kr
  - Shon, Hyoung Uk (손형욱) hyounguk.shon@kaist.ac.kr
  - Jeong, Woojin (정우진) woojin05@kaist.ac.kr

  - TA office hour: to be announced

# Instructor: Young-Gyu Yoon

- B.S. KAIST
- M.S. KAIST
  - Advised by Prof. SeongHwan Cho
  - Mixed-signal circuit (analog-to-digital converter design)
- Ph.D. MIT
  - Advised by Prof. Ed Boyden
  - Neuro-engineering (brain imaging & brain image analysis)

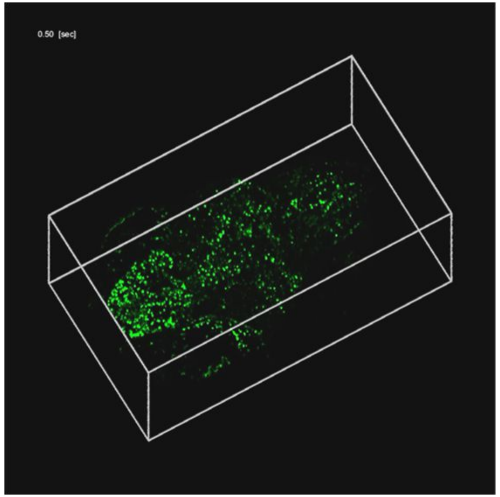
- Research engineer @ KAIST Institute
- Postdoc @ MIT
- Assistant professor @ KAIST

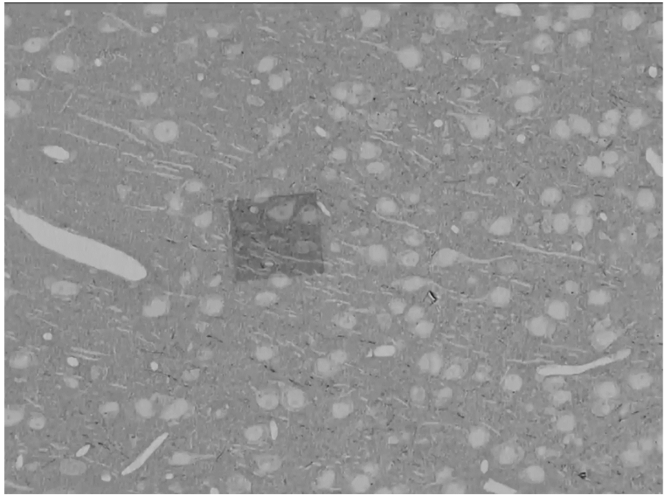
# Instructor: Young-Gyu Yoon

- Principal investigator of  Neuro-Instrumentation and Computational Analysis (NICA) Lab
  - Develop technologies for reverse-engineering a brain as a circuit

Neuro-Instrumentation & Computational Analysis Lab

- Research Area



# Contents

- Course logistics
- Course overview
  - Introduction to circuit engineering
  - Intended learning outcomes
  - Course topics

# Course logistics

- Class website
  - Basics of Artificial Intelligence<Big data analsys and machine learning> CoE202(A) @ [http://klms.kaist.ac.kr](http://klms.kaist.ac.kr)
  - Lecture notes & materials will be posted

- Class hours
  - 16:00AM – 17:30PM (Tue/Thu)

- Class room
  - Online lecture (zoom)

# Course logistics

- Midterm Exam
    - Two hours (Midterm )
    - Open book
    - No electronic device allowed

- Assignments
    - 3 programming assignments
    - Submit your iPython Notebook

- Final project
    - 1 final project (no final exam!)
    - Submit your report & iPython Notebook
    - Implementing your own neural network

# Course logistics

- Grading
  - Assignments (30%), Midterm exam (30%), Final exam (30%), Class participation (10%)
  - Grading ‘may’ be separately done for two groups
    - Group A: 3rd (or higher) year EE & CS students
    - Group B: other students

- Class participation
  - Attendance
  - In-class participation

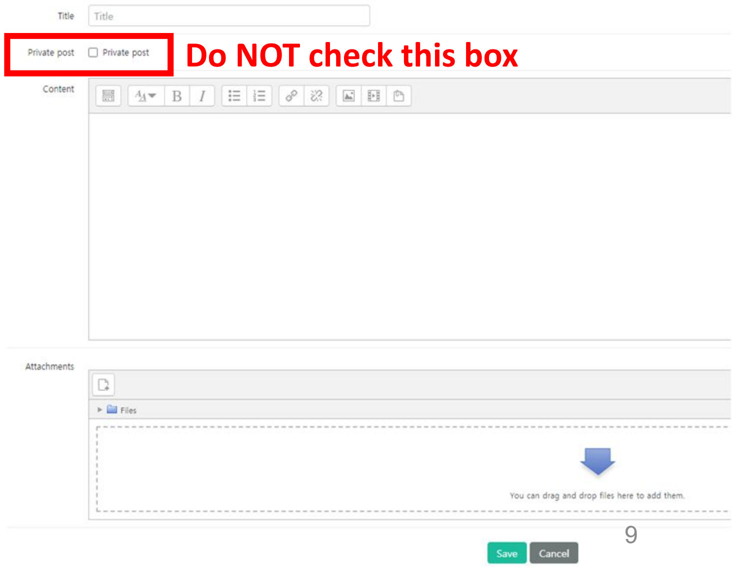
- Syllabus

# CoE202 Q&A

1. **Please Make your questions public, not private.**

2. **Please Use Q&A board @ KLMS, not e-mail.**

<u>We will not reply to private questions</u>
unless the question is 100%  personal (e.g., claim)




Do NOT check this box

# CoE202 Q&A rules

1. 1Please follow simple form for Title.
    1. **[homework name, code name, Lecture name] short summary**
        1. [HW1] typo at load mat function
        2. [note06.ipynb] Model does not converge
        3. [Week 5] Backpropagation implementation
2. Specify well. Upload your code snippet, environment and results so that we can help you efficiently.
3. Check others' question. You can get answer or hint. Please do not make duplicate question.

# 전기 및 전자공학부 학업 윤리 규정 제정 시행

목적

- 올바른 학업 윤리의식 함양 및 건전한 면학 분위기 확립
- 윤리위반행위에 대한 경각심 고취를 통한 윤리위반행위 사전 예방
- 윤리위반행위 발생 시 체계적인 대응 및 재발방지

적용

- 각 수업에서 허용하는 수업자료의 공유 및 참고 정책 사전 공지
- 정책 위반 행위 발생 시 규정에 명시된 절차에 따라 EE 학생 윤리 위원회에서 관련 조사/ 심의/ 학생지도/ 징계 진행

※ EE 홈페이지 공지사항에 학업 윤리 규정 전문 게시 https://ee.kaist.ac.kr/node/15358

학업 윤리 위반 행위 관련 신고/문의처 eehonor@kaist.ac.kr

정책공지

| **수업자료 공유정책**<br>(현 학기 중 공유/이후 학기 공유를 모두 포함) | 가능여부 (o/x) | 비고*<br>(수업 별 특수 정책이 있는 경우 명시) |
|---|---|---|
| 출제된 과제 및 과제풀이의 공유 및 배포 | | |
| 제공된 강의 자료의 공유 및 배포 | | |
| 출제된 시험문제의 공유 및 배포 | | |

| **이전 수업자료 참고정책** | 가능여부 (o/x) | 비고*<br>(수업 별 특수 정책이 있는 경우 명시) |
|---|---|---|
| 과제 수행 시 이전 기출 과제 및 과제풀이 참조 | | |
| 과제 수행 시 수강생 간의 토론/협업 | | |
| 시험 준비 시, 이전 기출문제 자료 참조 | | |

# 학업윤리 위반 사례 공유

| 위반행위 | 처분내용 | 발생 횟수 | 비고 |
|---|---|---|---|
| 과제물 표절 | 학부 내 징계<br>사회봉사 30시간 | 14 | |
| 허용되지 않은 자료<br>제공 및 전달 | 학부 내 징계<br>사회봉사 15시간 | 9 | |
| 부정행위 방조 및<br>허용되지 않은 자료 단순 열람 | 경고 (경고메일 발송) | 3 | |
| 튜터의 윤리위반 | 학부 내 징계<br>사회봉사 30시간, | 2 | 튜터링 과정에서 튜티에게 부정행위를 권유<br>및 지원한 경우 |
| 시험부정행위 | 유기정학 2개월,<br>사회봉사활동 100시간 | 2 | 전기및전자공학부 과목에서 발생한 부정행위<br>EE학생 윤리규정에 따라 처분. |
| | 학부 내 징계<br>사회봉사 60시간 | 1 | 타과 과목에서 발생한 부정행위, 과목 담당 교<br>수가 학생 징계를 요청하지 않은 케이스 |
| 시험부정행위<br>(익명제보) | - | 1 | 진상조사/면담/심의를 진행하였으나,<br>제보내용 외 부정행위를 확정할 수 있는 근거<br>가 없어 처분하지 않음. |

# Policies

**Sharing distributed material (present semester and after)**

| | Permission | Note* |
|---|---|---|
| **Assignments & solutions** | O* | *only private sharing allowed (i.e., do not upload on internet) |
| **Lecture material** | O* | *only private sharing allowed (i.e., do not upload on internet) |
| **Exams questions** | O* | *only private sharing allowed (i.e., do not upload on internet) |

**Reference material**

| | Permission | Note* |
|---|---|---|
| **Previous assignments & solutions** | O | |
| **Discussion among students** | O* | *Students are encouraged to discuss |
| **Previous Exams** | O | |

https://ee.kaist.ac.kr/en/node/15401

# Required background knowledge

- The courses assumes that the students are familiar with the “basics” of
  - Calculus

$$\nabla f(x,y,z) = \frac{\partial f}{\partial x}\mathbf{i} + \frac{\partial f}{\partial y}\mathbf{j} + \frac{\partial f}{\partial z}\mathbf{k} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{bmatrix}$$

  - Linear algebra

$$X^T Y = X^T X \theta \qquad \theta = (X^T X)^{-1} X^T Y$$

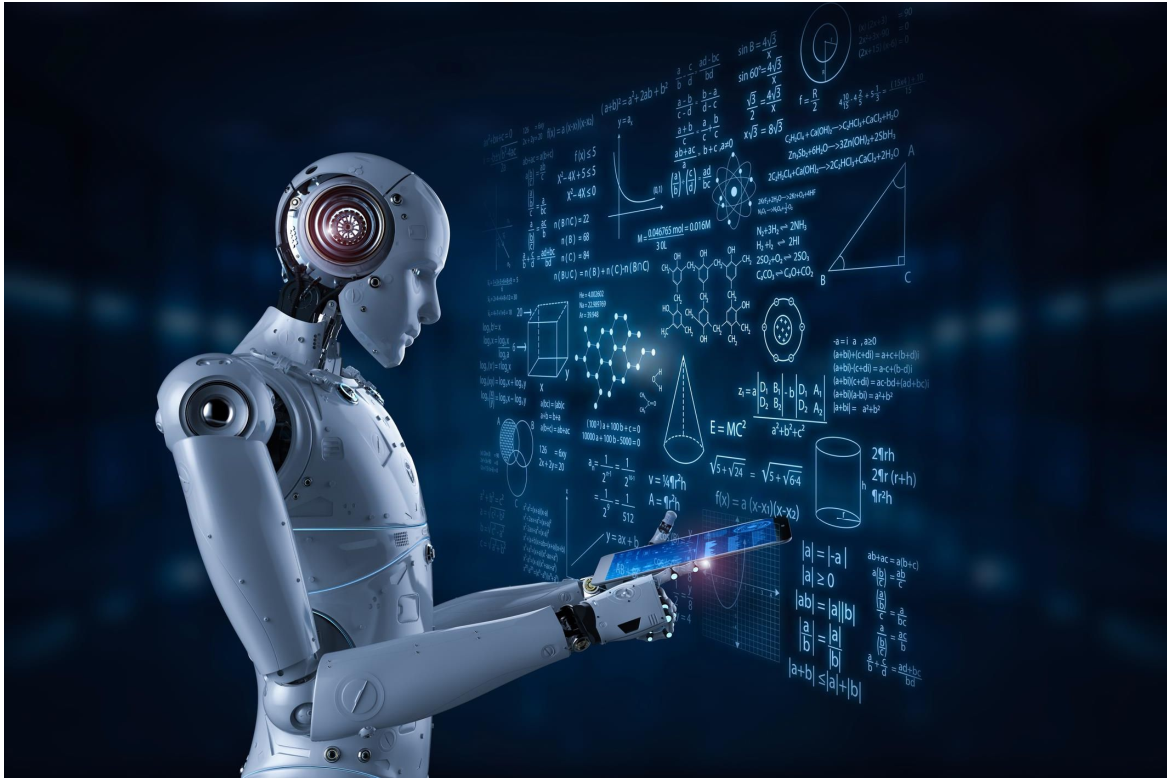
  - Probability

$$P(A|B)P(B) = P(B|A)P(A)$$

  - Python programming

```python
def quicksort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[len(arr) // 2]
    left = [x for x in arr if x < pivot]
    middle = [x for x in arr if x == pivot]
    right = [x for x in arr if x > pivot]
    return quicksort(left) + middle + quicksort(right)
```

# What is an artificial intelligence?



# Definition of machine learning

- A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

- Mitchell (1997)

**Algorithms that**

- **improve their performance**
- **at some task**
- **with experience**

# Types of machine learning

- **Supervised learning**: learning a function that maps an input to an output based on example input-output pairs
- **Unsupervised learning**: looking for previously undetected patterns in a data set with no pre-existing labels and without human supervision
- **Reinforcement learning**: enabling an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences

# Types of machine learning

- **Supervised learning**: “I’ll give you some pairs of questions and answers. Learn from these pairs to be able to answer to other questions.”
    - Regression
    - Classification

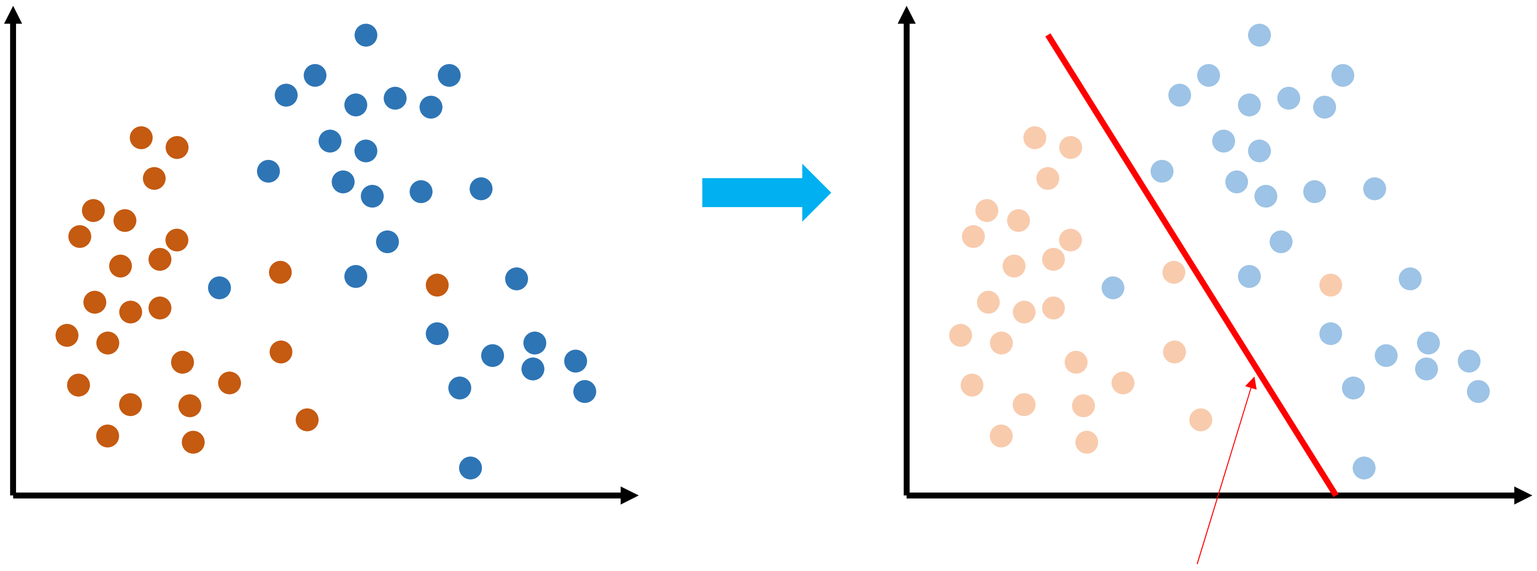
- **Unsupervised learning**: “I’ll give you some unlabeled data. Try to find if there’s any interesting structure or pattern in the data.”
    - Clustering
    - Dimension reduction

- **Reinforcement learning**: “I cannot teach you what to do, but I can give scores to what you did. Based on the scores you got from what you did, learn what to do.”

# Supervised learning

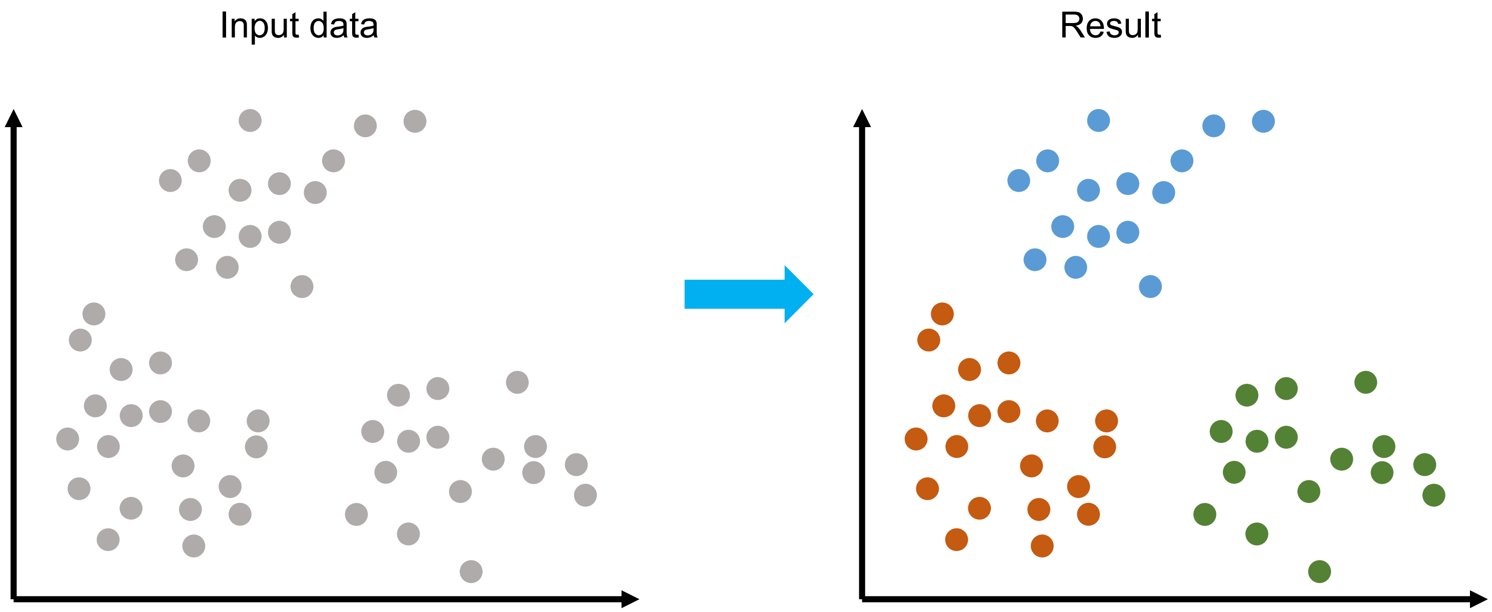
Training data

Classifier

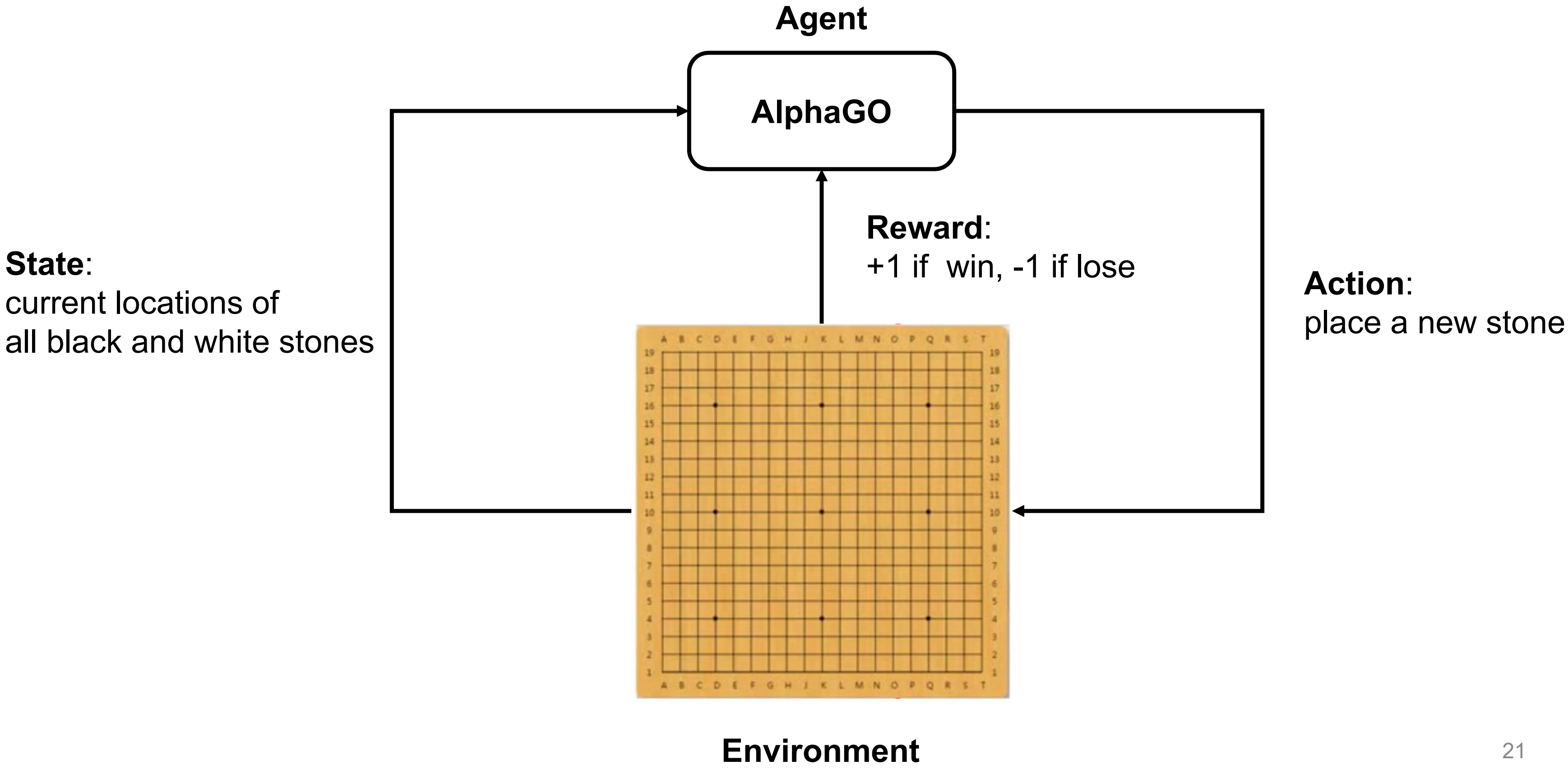


a rule that can classify unseen data

# Unsupervised learning



# Reinforcement learning

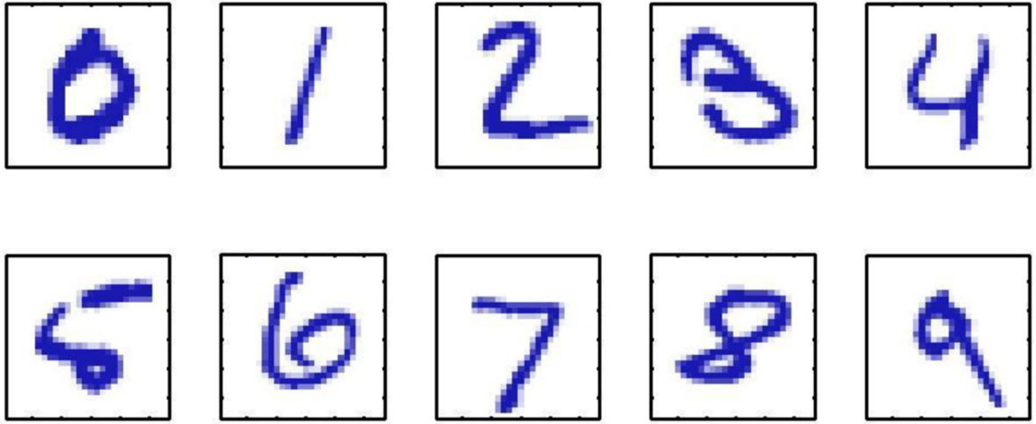


# The goal of this course…

- **Instructor's goal**
  - is to convince you that <u>AI is not difficult</u>
  - is to convince you that AI is not different than <u>polynomial curve fitting</u>
  - is to provide the basics and fundamentals of AI

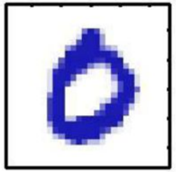
- **Student's goal**
  - is to understand the basics and fundamentals of AI
  - is to be able to design a simple AI

# Handwritten Digit Recognition

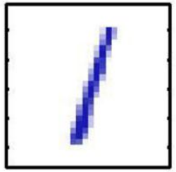


- Let's say we want to implement a program that can recognize handwritten digits. How can we design such program?

# Handwritten Digit Recognition



Design an algorithm that can detect a circular shape (but how?)

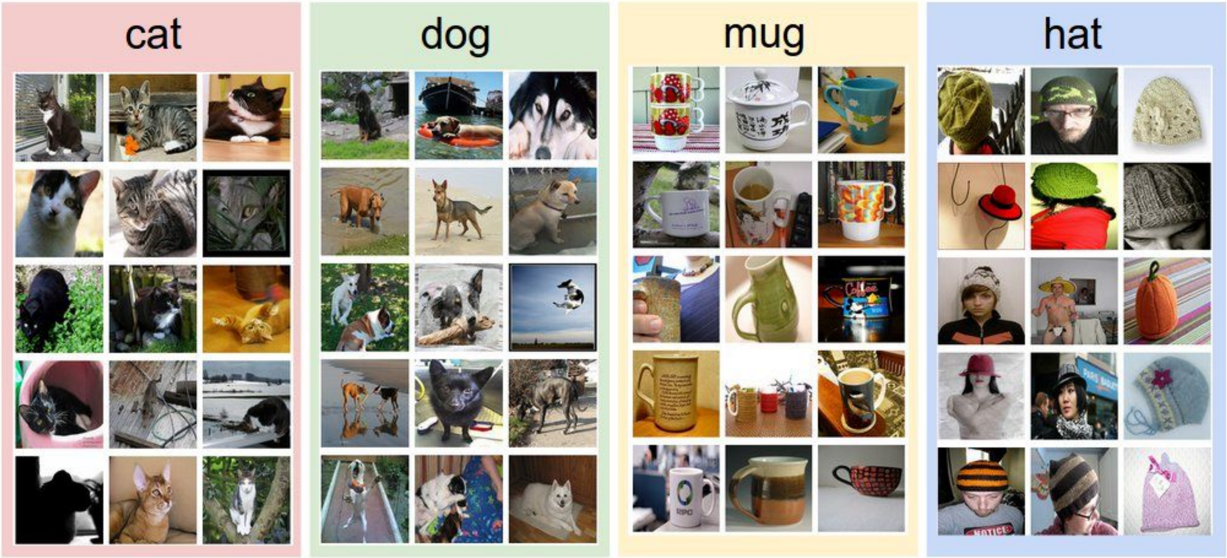


Design an algorithm that can detect a vertical line

Design an algorithm that can …

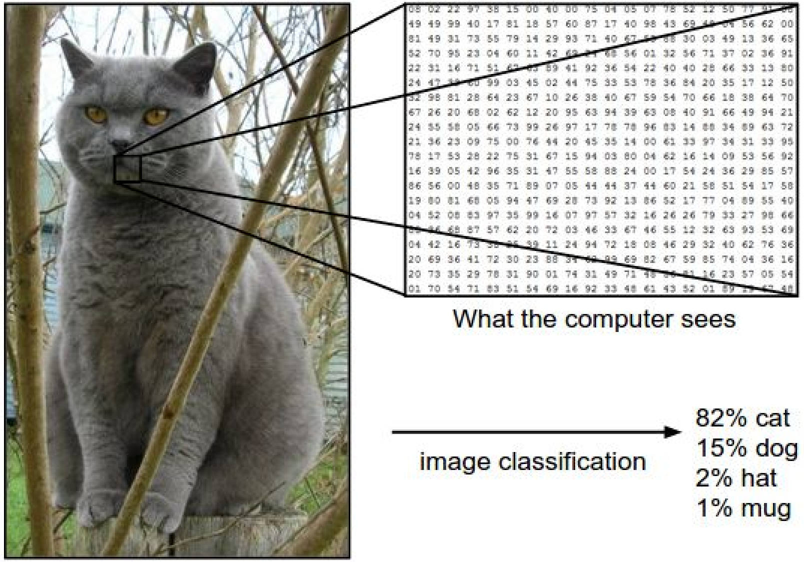
- And…remember that these are handwritten
  - Everyone has different handwriting

# Image Classification




- Let's say we want to implement a program that can classify images. How can we design such program?

# Image Classification




- For example, an image classification algorithm will take a numerical array (512 x 512 x 3) as the input, and generate another array (10x1) as the output



Figure from cs231n

# We need a measure of “similarity”

- Theses images do look visually similar to human

# We need a measure of “similarity”

$$\begin{pmatrix} 111 & 221 & 56 & 124 & 198 & 129 & 15 & 91 & 198 \\ 12 & 232 & 236 & 134 & 91 & 99 & 32 & 143 & 108 \\ 16 & 223 & 81 & 55 & 146 & 15 & 16 & 72 & 211 \\ 116 & 53 & 31 & 77 & 177 & 163 & 16 & 12 & 22 \\ 36 & 23 & 131 & 87 & 64 & 99 & 46 & 176 & 143 \\ 116 & 222 & 85 & 44 & 126 & 9 & 6 & 192 & 197 \\ 255 & 123 & 123 & 124 & 77 & 19 & 156 & 82 & 211 \\ 122 & 200 & 203 & 24 & 21 & 18 & 191 & 199 & 1 \end{pmatrix} \qquad \begin{pmatrix} 44 & 221 & 26 & 34 & 178 & 19 & 61 & 91 & 98 \\ 12 & 232 & 136 & 134 & 91 & 99 & 32 & 53 & 18 \\ 16 & 43 & 93 & 105 & 66 & 15 & 16 & 72 & 3 \\ 116 & 53 & 4 & 77 & 14 & 93 & 71 & 33 & 2 \\ 36 & 23 & 41 & 157 & 64 & 18 & 200 & 176 & 3 \\ 26 & 222 & 85 & 44 & 126 & 9 & 6 & 192 & 7 \\ 65 & 123 & 123 & 124 & 177 & 19 & 156 & 42 & 121 \\ 32 & 100 & 103 & 124 & 121 & 218 & 11 & 99 & 1 \end{pmatrix}$$

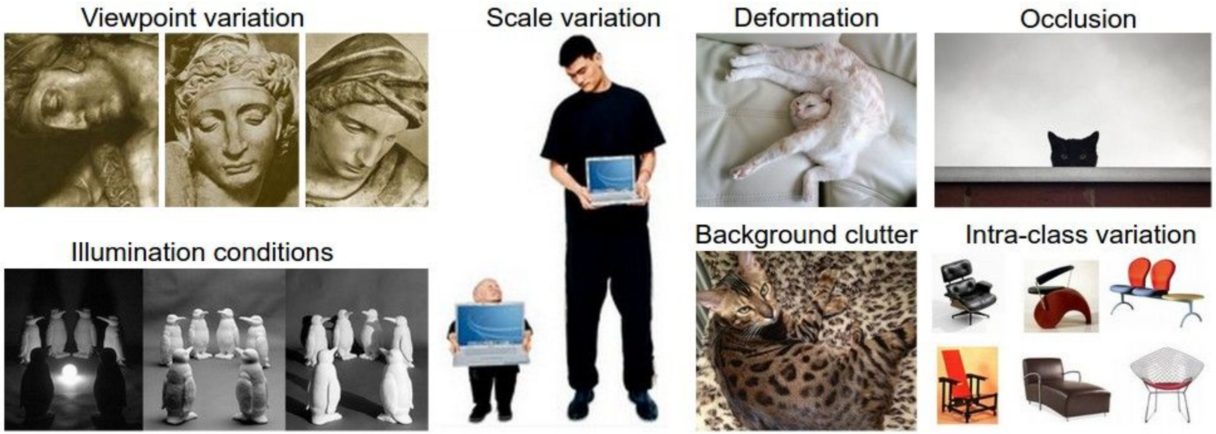
- But, will these matrices look *similar to the computer?
- How can we define the similarity of two different matrices for image classification?

*again, how can we define similarity?

# Challenges in image classification

- **Viewpoint variation**. A single instance of an object can be oriented in many ways with respect to the camera
- **Scale variation**. Visual classes often exhibit variation in their size (size in the real world, not only in terms of their extent in the image)
- **Deformation**. Many objects of interest are not rigid bodies and can be deformed in extreme ways
- **Occlusion**. The objects of interest can be occluded. Sometimes only a small portion of an object (as little as few pixels) could be visible
- **Illumination conditions**. The effects of illumination are drastic on the pixel level
- **Background clutter**. The objects of interest may *blend* into their environment, making them hard to identify
- **Intra-class variation**. The classes of interest can often be relatively broad, such as *chair*. There are many different types of these objects, each with their own appearance

# Challenges in image classification




- **Viewpoint variation**
- **Scale variation**
- **Deformation**
- **Occlusion**
- **Illumination conditions**
- **Background clutter**
- **Intra-class variation**

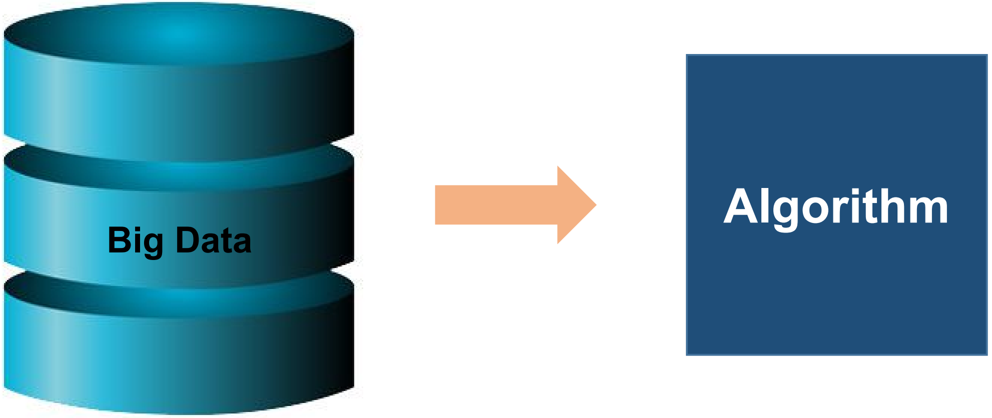
# Question






- Despite all these issues, we (human) have no problem in recognizing that these are cats
- How can our algorithms do the same?

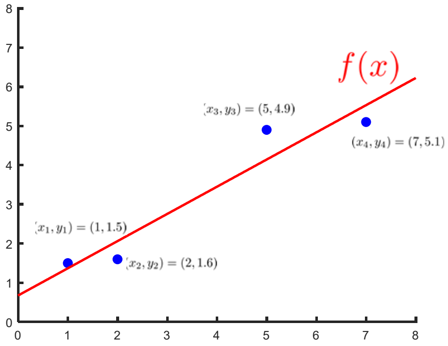
# Data-driven approach




- What if we can **design a program** that can analyze the data and make its own algorithm?
  - Give all possible variations (viewpoint, scale, etc) and just let the program make the algorithm

# Revisit: Goal

## Regression




$$f(x) = 0.6934x + 0.6747$$

$\approx$

## Classification

CAT

# Course overlook

- Supervised Learning
  - Regression
    - Linear fitting
    - Polynomial fitting
  - Classification
    - Linear classification
    - Nonlinear classification (neural network)
- Unsupervised learning
  - Clustering
  - Autoencoder
- Reinforcement learning

# Class organization

- Class types
  - Lecture
    - Focus on theoretical aspects
  - Activity
    - Hands-on practice (python programming using Google Colab)
      - The instructor will show you an example
      - A task, similar to the example, will be given to students
- The purpose is to
  - Learn fundamental theories
  - Experience basic machine learning
  - Bridge the gap between theory and practice

# Summary

- Course logistics
- Definition of artificial intelligence
  - Supervised learning
  - Unsupervised learning
  - Reinforcement learning
- Image classification
  - Challenges in image classification
- Data driven approach

# References

- **Lecture notes**
  - **MIT 6.036 Intro to Machine Learning**
    - https://www.mit.edu/~lindrew/6.036.pdf
  - Stanford CS229
    - http://cs229.stanford.edu/syllabus-summer2020.html

- Website
  - CS231n course website: https://cs231n.github.io/