Introduction to Machine Learning

Lecturer: Cheng-Chin Chiang

Department of CS & IE

National Dong Hwa University

Books

- No textbook
- Reference books:
    - *Pattern Recognition and Machine Learning*, by Christopher M. Bishop
    - *Introduction to Machine Learning*, by Ethem Alpaydin
    - Learning from Data, by Yaser Abu-Mostafa, Malik Magdon-Ismail and Hsuan-Tien Lin
- Lecture Notes:
    - Mostly made by Cheng-Chin Chiang
    - Partly revised from Prof. Hong-Yi Lee (Prof. of EE Department of NTU)

Grade Evaluation

- Three programming assignments (75%)
- One midterm exam (25%)
- Two roll calls for bonus (10%)
  - may or may not happen depending on the class attendance of the class day

Important Note 1

- As this course will be a programming-intensive course, please get prepared on your programming skills.
  - **Python** is powerful because
    - it has fruitful **Machine Learning packages**
  - **Python** is potential because
    - it is getting more and more popular in **AI and ML research and development**.
  - **Python** is easy to learn because
    - Its syntax is much more natural than **C, C++, and Java**.
  - Python is economical because
    - It is **free**, unlike the MATLAB which costs much.

Important Note 2

- All programs written for this course must be well version-controlled through **GitHub**.
  - Version control is a basic skill for any software engineer. Learn it before you become a professional programmer.
  - Version control gives your programs a good development log.
  - Version control can prevent your work from unexpected loss or damages of program codes.
  - Version control **unveils the plagiarism and show how much effort you've devoted to**.
    - **Any plagiarism behavior will be punished by grading with E in the final credit evaluation.**

The topics to be covered

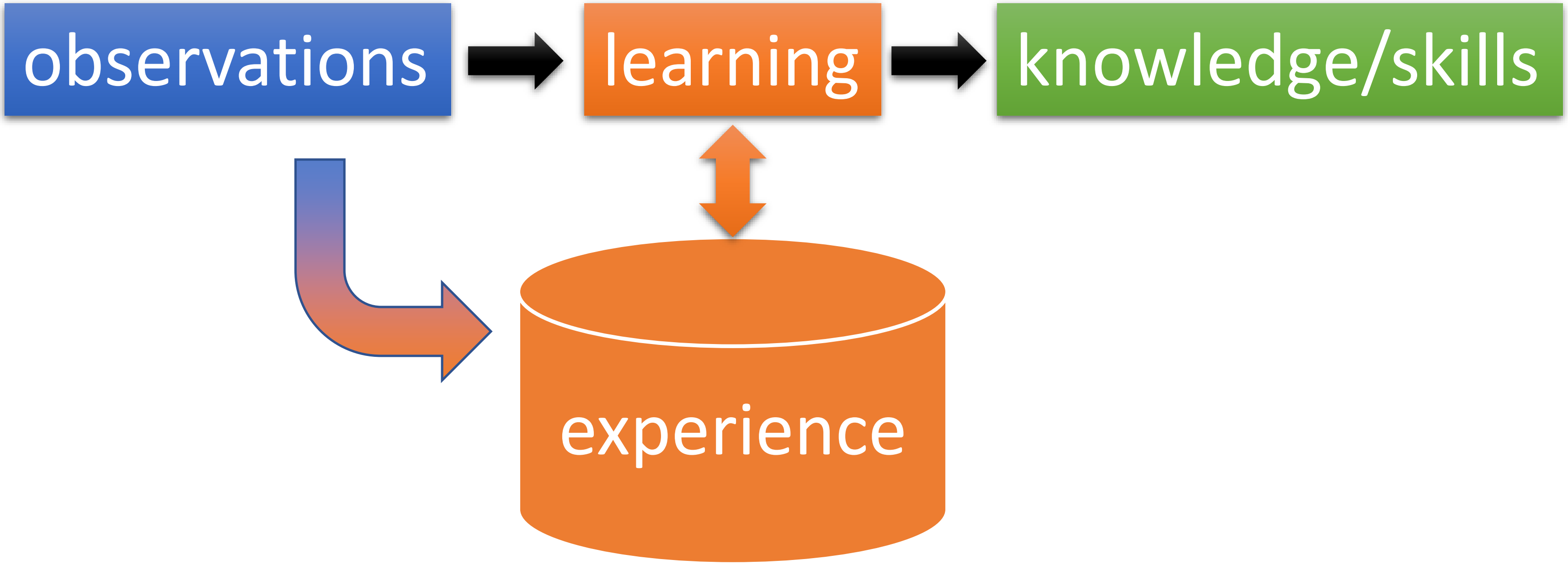
- K-Nearest Neighbor Classifiers
- Naïve Bayes Classifiers
- Linear Classification
- Logistic Classification
- Support Vector Machines
- Decision Trees
- Ensemble Methods
- Neural Networks

Background needed

- Math
  - Basic Calculus
  - Linear Algebra
  - Probability

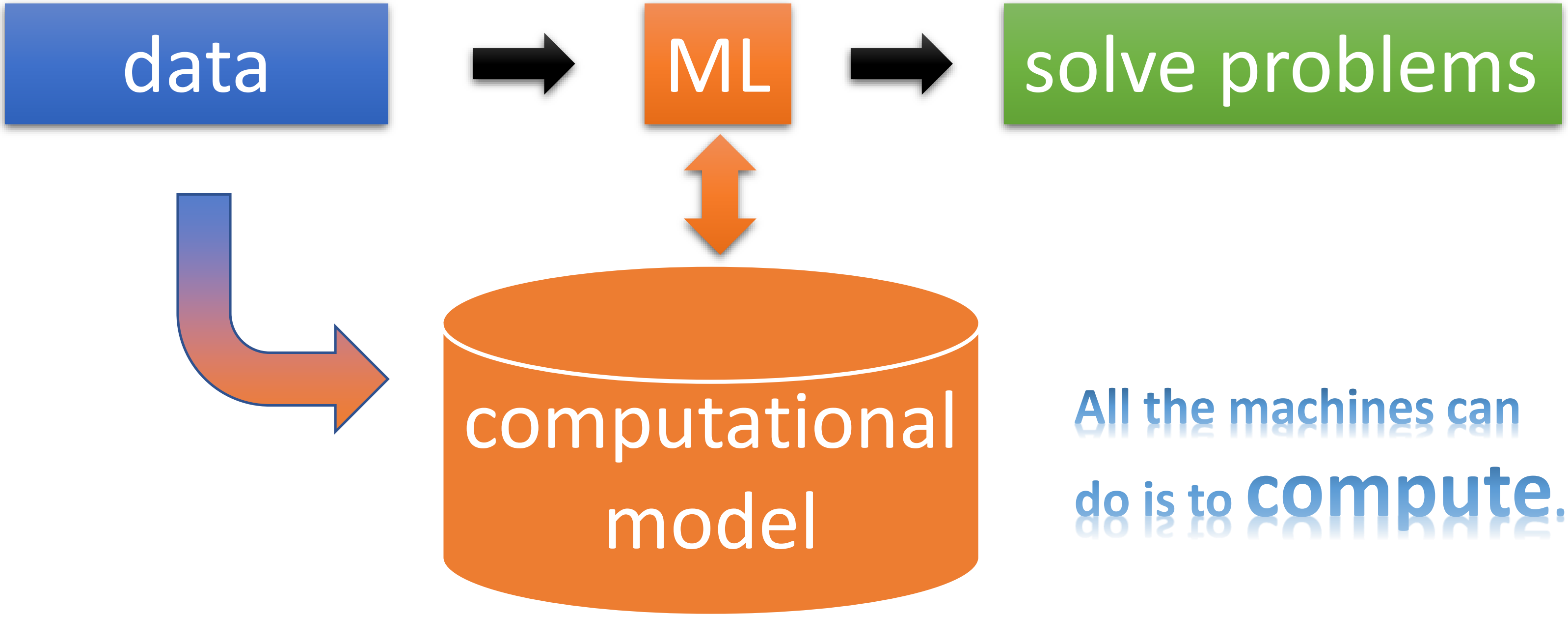
Human Learning

- Goal: **acquiring knowledge/improving skills**
- How: **experience** from **observations**



Machine Learning (ML)

- Goal: **solving problems**
- How: **computational model** from **collected data**



All the machines can do is to **compute**.

Why ML?

- cost-effective decision   low-cost and high efficacy
  - product inspection:
    - human worker can get exhausted and make mistakes accidently
  - customer financial crediting:
    - computerized clients' crediting records are more reliable than subjective human judgement on making loans

- beyond human ability   high storage and fast computing
  - board games
    - The number of possible configurations of Go are so large that a human player cannot forecast.
    - Recall of human memories is often slow, incomplete and unreliable.

When ML?

- A lot of data is available.
  - Using 1,000,000 images to train an object recognizer.
- Some underlying patterns exist.
  - Guessing a random number has no patterns.
  - Predicting the country-wide electrical load in a day has some patterns, e.g., getting higher at noon.
- No exact solution is available.
  - Don't apply ML algorithms to any well-solved problems.
    - A quicksort algorithm can outperform all existing machine learning algorithms
    - ML cannot better solve the shortest path problem than the Dijkstra's algorithm,
  - However, ML plays Go better than a world champion player.

Key Concerns

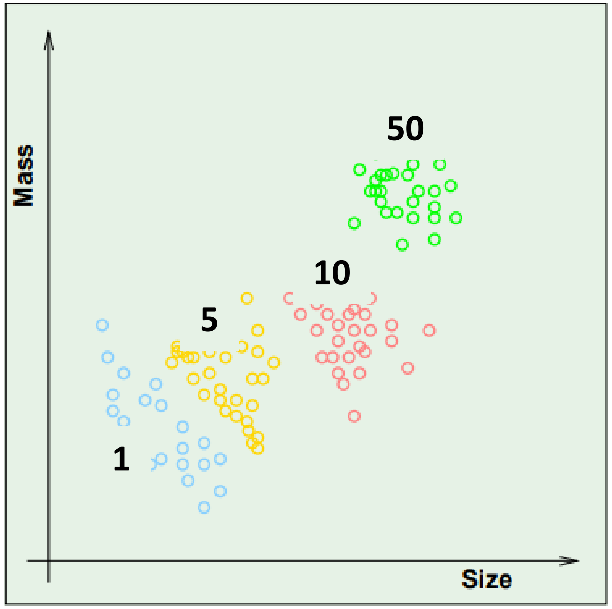
- Task Type?
  - regression?
  - classification?
  - clustering?
- Data?
  - features?
  - transformations?
  - labels?
- Performance?
  - how to measure?

Example Applications

- **Person identification**
  - Task type: classification
  - Data: face image
  - Performance: recognition rate (percentage)
- **Temperature prediction**
  - Task type: regression
  - Data: temperatures of the past week
  - Performance: accuracy (degree)
- **Document categorization**
  - Task type: clustering
  - Data: keywords, titles
  - Performance: category compactness

Formalizing a Learning Problem

- **Coin Recognition**



Data: $\mathbf{X} = \{\mathbf{x}_i = [m_i, s_i]\}_{i=1}^{N}$

Output: $Y = \{y_i \in \{1,2,3,4\}\}_{i=1}^{N}$

Target Function (unknown):

$$\boldsymbol{f}: \mathbf{X} \rightarrow Y$$

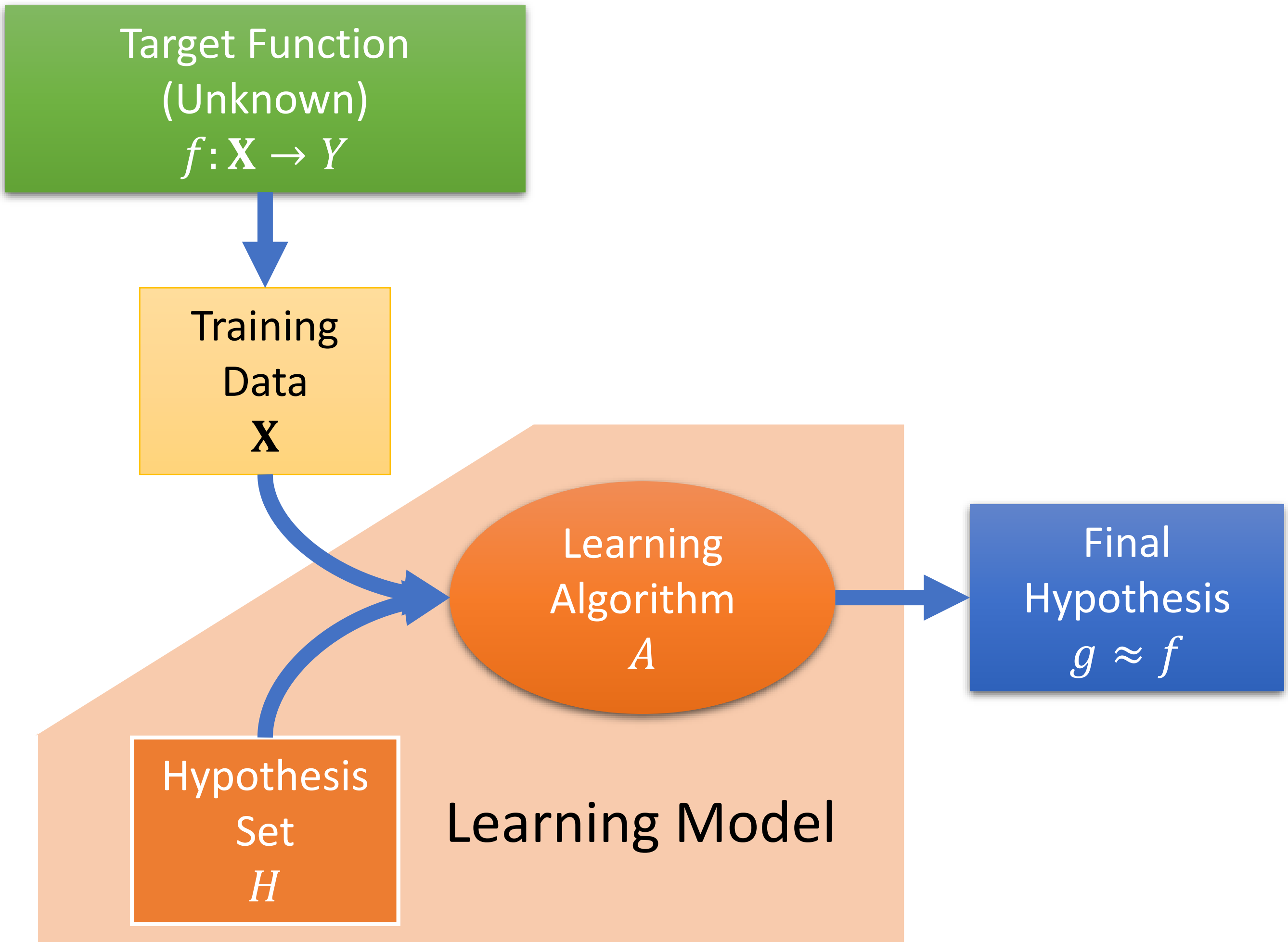
Hypothesis:

$$\boldsymbol{H}: \mathbf{X} \rightarrow Y$$

Algorithm: picks g from H to approach f

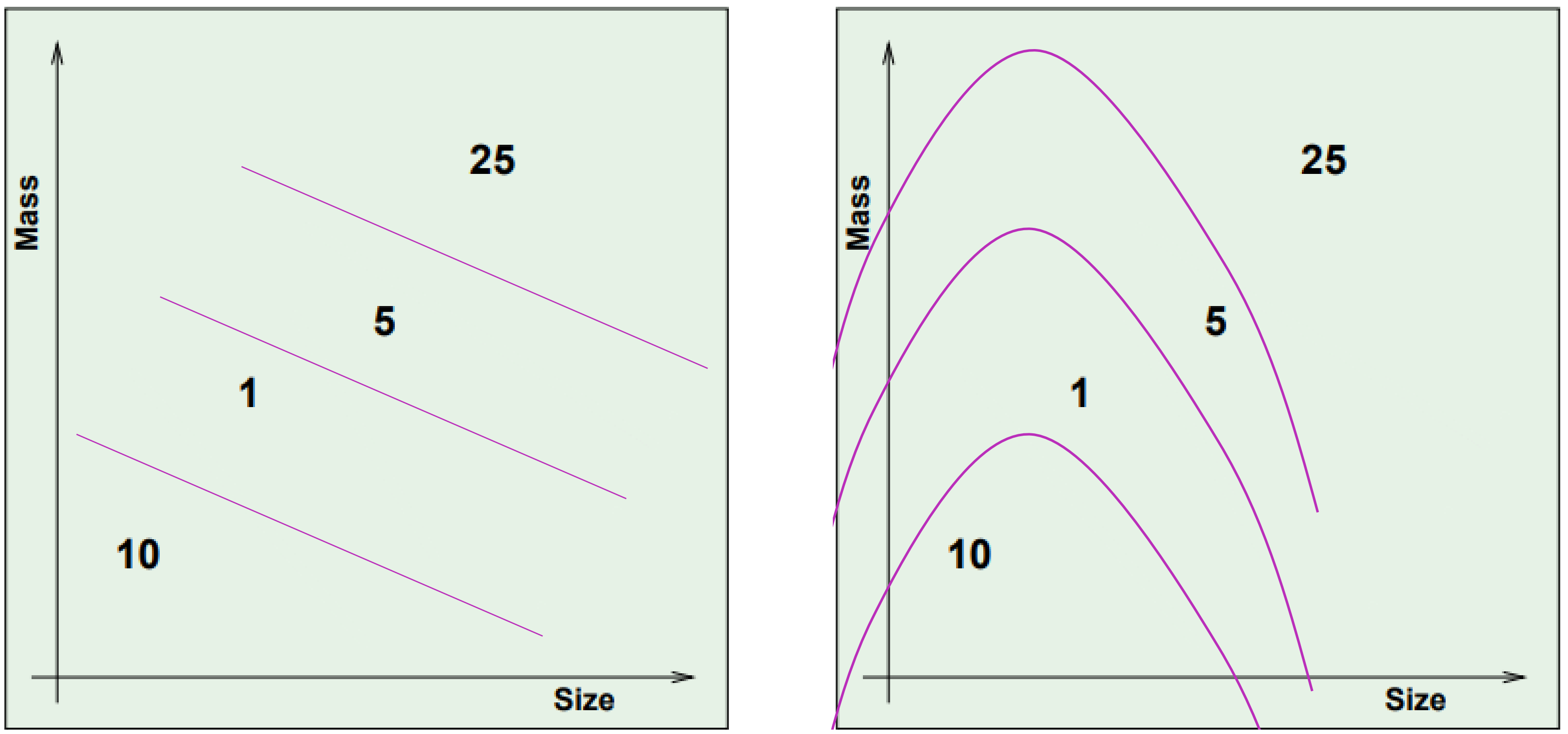
$$A: H \rightarrow g \approx f$$

Formalizing a Learning Problem (cntd.)



Types of Hypotheses

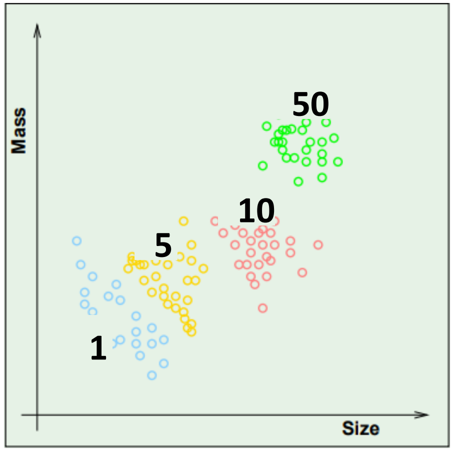
- **Linear**
  - Assume g to be a linear function of inputs.
  - $Ex.:$
    - $$g(size, mass) = \begin{cases} 1, & size * \mathbf{2} + mass * \mathbf{1} + \mathbf{1} \leq \mathbf{1} \\ 2, & \mathbf{1} < size * \mathbf{2} + mass * \mathbf{1} + \mathbf{1} \leq \mathbf{2} \\ 3, & \mathbf{2} < size * \mathbf{2} + mass * \mathbf{1} + \mathbf{1} \leq \mathbf{3} \\ 4, & \mathbf{3} < size * \mathbf{2} + mass * \mathbf{1} + \mathbf{1} \end{cases}$$
- **Nonlinear**
  - Assume g to be a non-linear function of inputs.
    - $$g(size, mass) = \begin{cases} 1, & \mathbf{-1} * size^2 + \mathbf{1} * mass + \mathbf{1} \leq \mathbf{1} \\ 2, & \mathbf{1} < \mathbf{-1} * size^2 + \mathbf{1} * mass + \mathbf{1} \leq \mathbf{2} \\ 3, & \mathbf{2} < \mathbf{-1} * size^2 + \mathbf{1} * mass + \mathbf{1} \leq \mathbf{3} \\ 4, & \mathbf{3} < \mathbf{-1} * size^2 + \mathbf{1} * mass + \mathbf{1} \end{cases}$$



Example of Coin Recognition

Types of Learning

- Supervised Learning
  - Data = (Input, Correct Output)
- Unsupervised Learning
  - Data = (Input)
- Reinforcement Learning
  - Data = (State, Action, Reward)

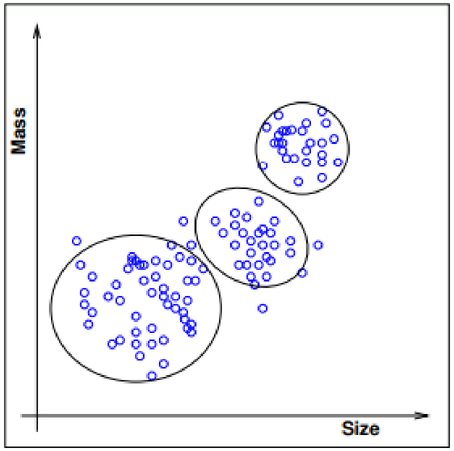


Supervised Learning
(Regression/Classification)

I1, O1 => adjust system parameter

I2, O2 => adjust system parameter

In, On => adjust system parameter



Unsupervised Learning
(Clustering)

I1 => adjust system parameter

I2 => adjust system parameter

In => adjust system parameter

Reinforcement Learning
(Learning from rewarded actions)

I1 => action 1➔reward 1➔adjust

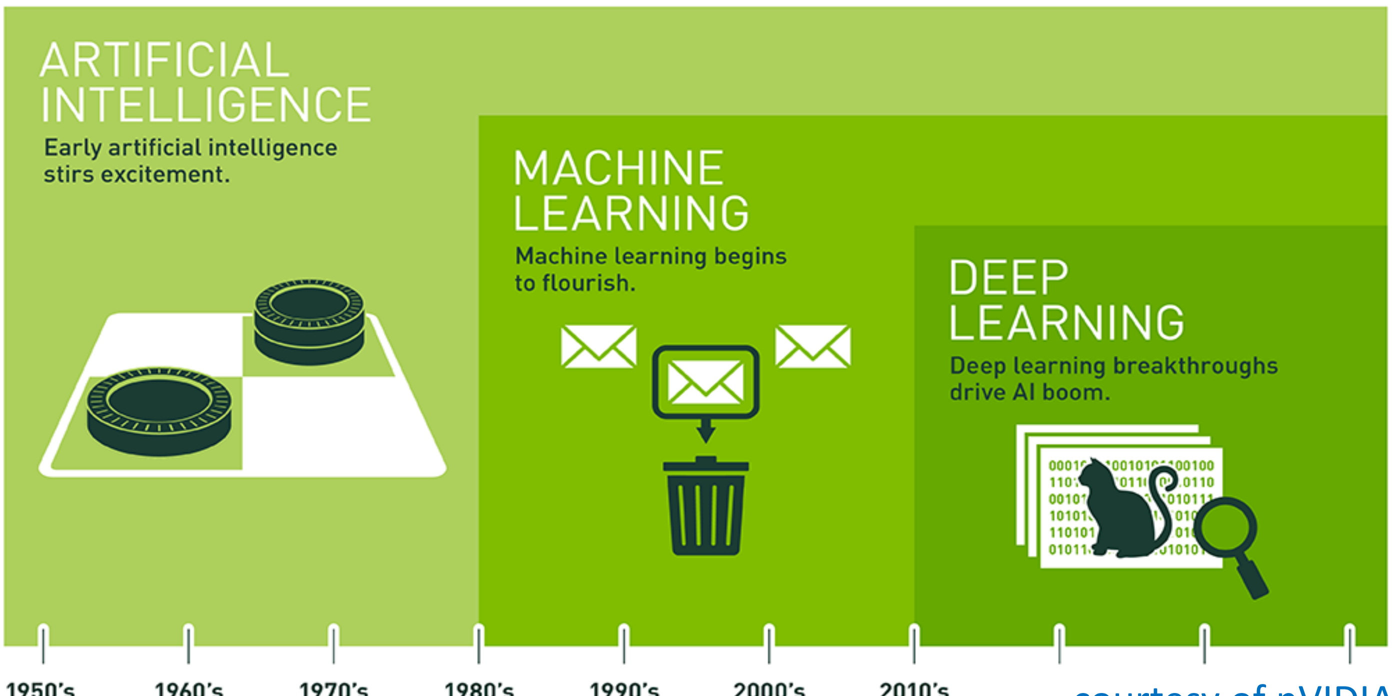
I2 => action 2➔reward 2➔adjust

In => action n ➔reward n➔adjust

Artificial Intelligence, Machine Learning, and Deep Learning

ML is one approach to realize AI,

while DL is one technique to realize ML.



courtesy of nVIDIA

Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.