Introduction of Artificial Intelligence for UAS

AAE4011 – Artificial Intelligence for Unmanned Autonomous Systems (UAS)

Dr Weisong Wen
Assistant Professor, Director of PolyU TAS Lab
Department of Aeronautical and Aviation Engineering
The Hong Kong Polytechnic University
Week 1, S2, 2024/2025

Department of Aeronautical and Aviation Engineering 航空及民航工程學系

THE HONG KONG POLYTECHNIC UNIVERSITY 香港理工大學

Dr Weisong Wen (Trustworthy Autonomous Systems)

Sept 2023-Now, Assistant Professor
2021-Aug 2023, Research Assistant Professor
2020-2021, Postdoc at AAE
2017-2020, PhD from Mechanical Engineering

2018, Visiting Researcher in UC Berkeley

2016-2017, Co-found Researcher in iDriverplus (estimated value RMB 5 billion)

2015, 2017, BSc, MSc of Mechanical Engineering from BISTU and CAU

Selected academic contributions

- **33 SCI journal (17 Q1, 11 Q2) and 40 conference papers** (highly cited papers, Best Presentation Award etc.)
- Session chair of ION GNSS+ 2022/2023, ICGNC 2022 and IEEE ITSC 2022/2023
- Over 19 millions HKD of research fund as PI

Funder (as PI since 2021)

創新科技署 Innovation and Technology Commission

RI-AM Research Institute for Advanced Manufacturing

广东省科学技术厅 Department of Science and Technology of Guangdong Province

美团 Meituan Tencent 腾讯 HUAWEI

UNITY DRIVE 一清创新

運輸署 Transport Department

RILS

Selected research collaborators

東京海洋大学 Tokyo University of Marine Science and Technology

TECHNISCHE UNIVERSITÄT CHEMNITZ

Leibniz Universität Hannover

Massachusetts Institute of Technology

ASTRI 香港應用科技研究院

Northeastern University

Cogniser 中科凯泽

RWTH AACHEN UNIVERSITY

DLR Deutsches Zentrum für Luft- und Raumfahrt German Aerospace Center

武漢大學 WUHAN UNIVERSITY

Our Teaching Assistants


Zhang Ziqi


Yang Qian


Wang Xin


Qiu Shaoting


Hu Runzhi

- Role/features of TAs in this course
  - Helper in lab session
  - Expert in AI and coding with Python
  - Experts in UAS, such as drones

- Let's get to know with each other
  - Short introduction about yourself (if we have enough time)? ☺
  - Who is your Final Year Project supervisor and what is your topic? ☺
  - Why you select this course? ☺

Ground Rules

✓ For students:

Open mind; speak English; participate activities assigned; ask questions

✓ For teachers:

Arrive on time; reply emails on time; answer questions related to the subject

✓ Be curious, Be inspired, Be motivated, Study further by yourself.

Assessment and Basic Requirement

- Assessment:
  - Homework Assignment (Strictly no late submission) (20%)
  - Mid-Term Quiz/Test (15%, close book)
  - Group Project (Case study, several members in a group) (15%)
  - Final Exam (50%) (Open book)
- Basic requirement:
  - Mathematics on matrix and its calculation
  - Extra time for finish the coding homework based on Python
  - Assurance on the attendance
  - Basic coding skills with Python (expect to learn yourself for extra), one week lecture for basics of Python

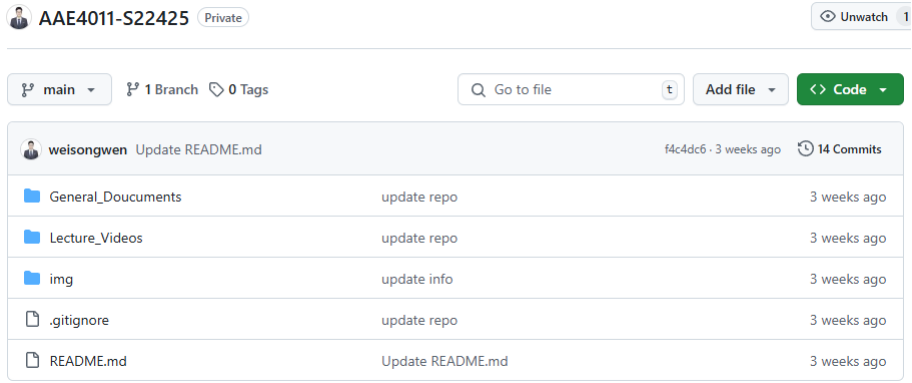
Collaborate with Github

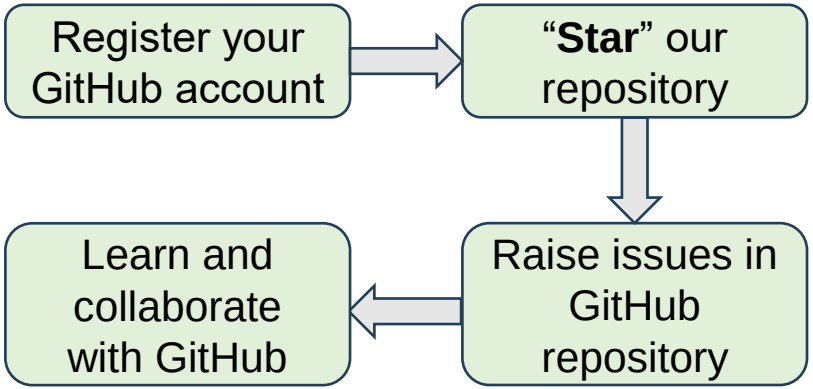
https://github.com/weisongwen/AAE4011-S22425

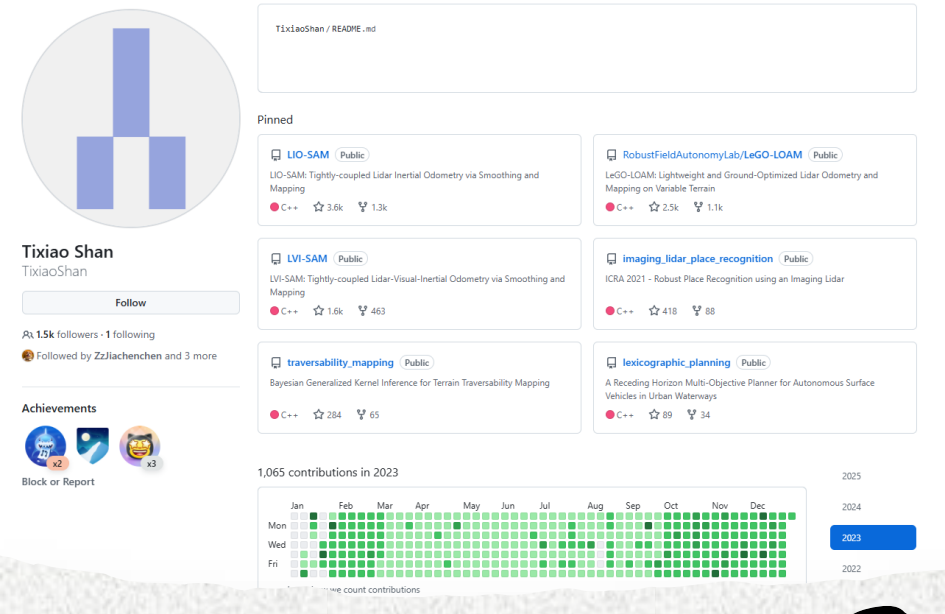
AAE4011, Artificial Intelligence for Unmanned Autonomous Systems

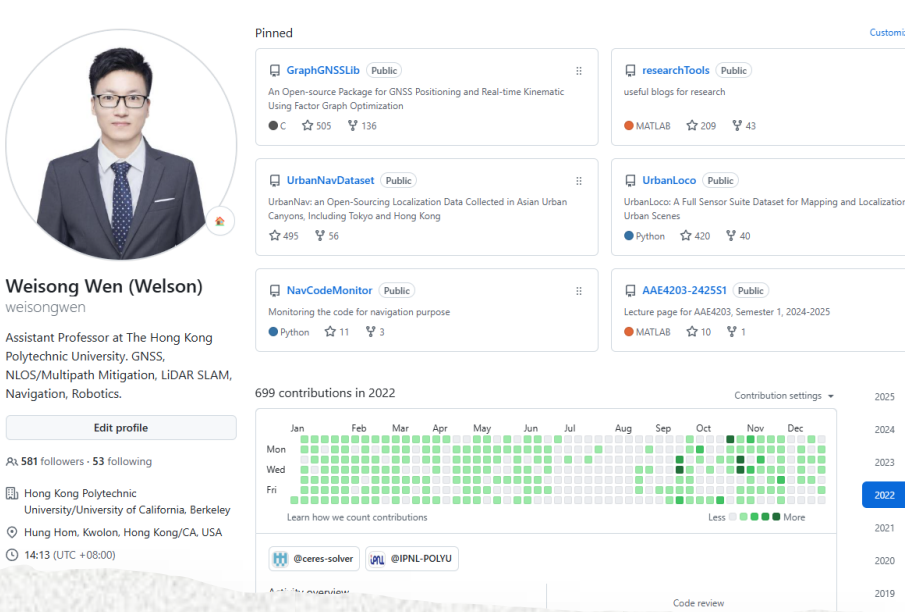
This is the Github page for lecture AAE4011, Artificial Intelligence for Unmanned Autonomous Systems. This page will include the example code, and some learning materials. More importantly, students can raise issues from this Github repository, through Issue item.

- **Lecturer:** Dr. Weisong Wen, Departmnet of Aeronautical and Aviation Engineering (AAE)
- **Adddress:** R820, PolyU.
- **Email:** welson.wen@polyu.edu.hk
- **Teaching Assistant:** Shaoting Qiu, Xin Wang.
- **Lecture Venue and Time:** FJ303, Fri 08:30-11:20 AM









GitHub can be part of your resume!

- How many popular repositories your have?
- How many followers you have?
- How active you are in GitHub?

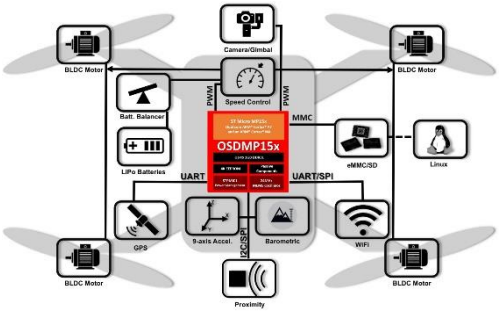
Teaching Plan (Tentative)

Sem. Week	Topics Taught	Assessment
Week1	Introduction of the Artificial Intelligence for Unmanned Autonomous Systems	Assignment 1
Week 2	Fundamentals of Programming for AI (Python) (**take your laptop**)	
Week 3	Introduction to the Unmanned Autonomous Systems (Drones)	
Week 4	Linear Regression and Logic Regression: Theory and Applications	
Week 5	Unsupervised Learning: Theory and Applications	
Week 6	Supervised Learning: Theory and Applications (part 1)	
Week 7	Lab Session (**take your laptop**)	Lab Report
Week 8	Mid-term	Assignment 2
Week 9	Supervised Learning: Theory and Applications (part 2)	
Week 10	Multiple Layer Perceptron and CNN: Part 1	
Week 11	No lectures	
Week 12	Case Study Presentation	Group Presentation & Report
Week 13	Case Study Presentation (Makeup class during the Weekend)	Group Presentation & Report

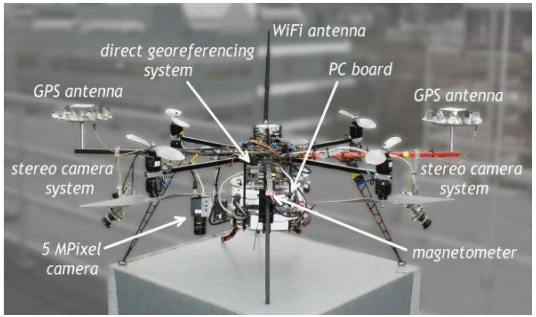
Intended Learning Outcomes

- Understand the applications, core concepts, characteristics and current state of Artificial Intelligence and unmanned autonomous systems .
- Proficiency in using AI implementations, specific techniques and tools in particular problems.
- Mastery of the ability to complete machine learning, deep learning tasks, and complete some demo codes.
- Summarize trends in AI technology and its application in high level intelligence.

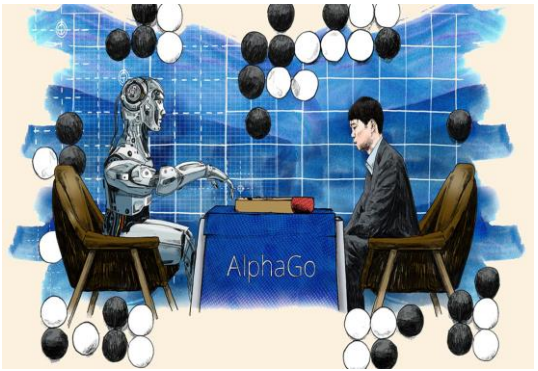
Components and system for drone [Source]



Sensors, perception system for drones [Source]



AI enabled detection for drones [Source]



AI for high level decision making [Source]

Outline

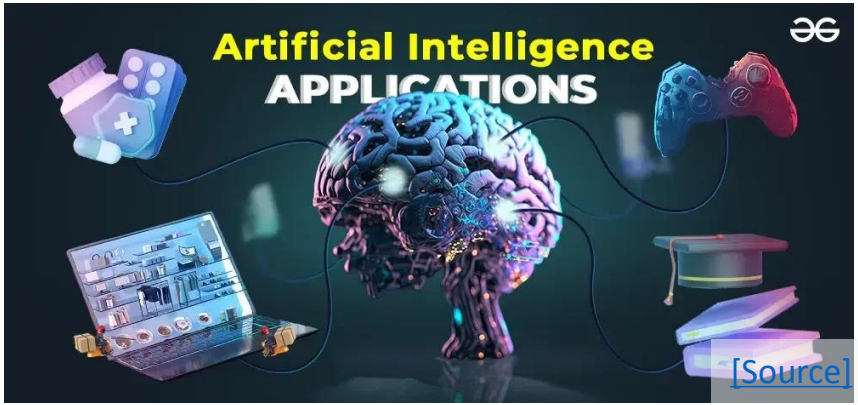
- AI concepts:
  - What is AI? Why need AI?
  - History of AI, Classification of AI technologies, Machine learning, Deep learning, Reinforce learning
  - Basic understanding of ROS
  - Q&A
- Applications of AI technology for UAS:
  - Common AI tools, frameworks
  - Specific techniques and tasks
  - Concepts for Regression, Classification, Clustering
  - Standard implementation process
  - Q&A

What is *Artificial intelligence (AI)*?

Artificial intelligence (AI), in its broadest sense, is intelligence exhibited by machines.

AI is empowering massive applications!

- Top 20 Applications of Artificial Intelligence (AI) in 2024
  - 1. Artificial Intelligence in E-Commerce
  - 2. Artificial Intelligence in in Education Purpose
  - 3. Artificial Intelligence in Robotics
  - 4. Artificial Intelligence in GPS and Navigations
  - 5. Artificial Intelligence in Healthcare
  - 6. Artificial Intelligence in Automobiles
  - 7. Artificial Intelligence in Agriculture
  - 8. Artificial Intelligence in Human Resource
  - 9. Artificial Intelligence in Lifestyle
  - 10. Artificial Intelligence in Social media
  - 11. Artificial Intelligence in Gaming
  - 12. Artificial Intelligence in Astronomy
  - 13. Artificial Intelligence in Chatbots
  - 14. Artificial Intelligence in Surveillance
  - 15. Artificial Intelligence in Finance
  - 16. Artificial Intelligence in Data Security
  - 17. Artificial Intelligence in Travel and Transport
  - 18. Artificial Intelligence in Marketing
  - 19. Artificial Intelligence in Entertainment
  - 20. Artificial Intelligence in Military

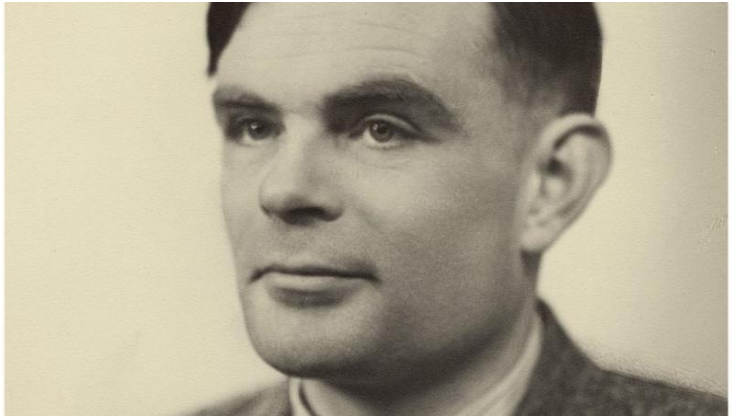


[Source]

AI enabled drones, ChatGPT for your learning (maybe one of your best friends…), drones, humanoid robots, self-driving cars…

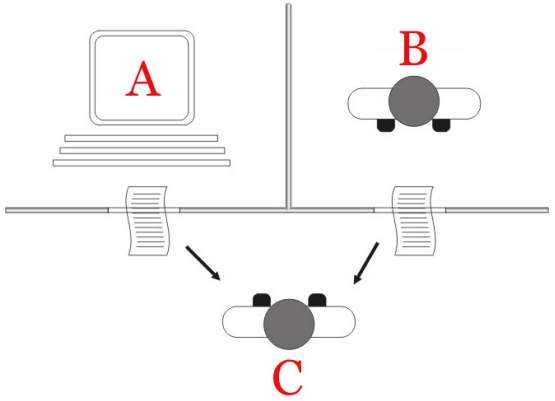
Can machines think?

The Intersection of Science Fiction and Reality: **Can machines think?**



Alan Turing, father of artificial intelligence

The **Turing test**, originally called the **imitation game** by Alan Turing in 1949, is a test of a machine's ability to exhibit intelligent behaviour equivalent to, or indistinguishable from, that of a human.



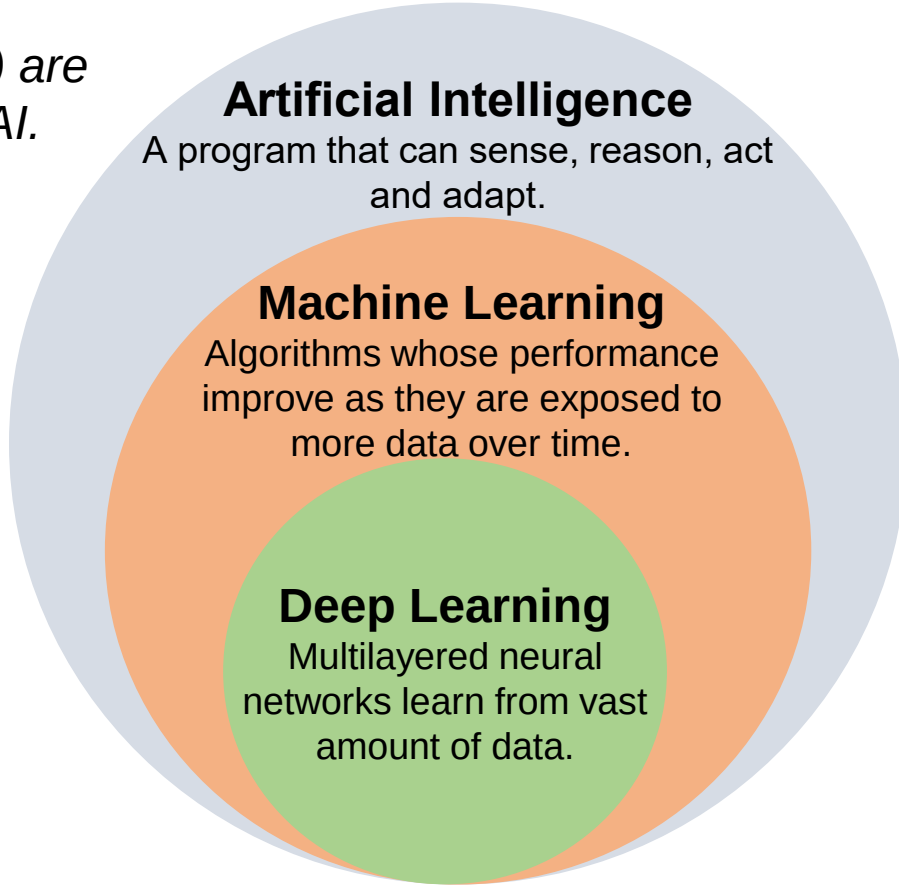
The "standard interpretation" of the Turing test, in which player C, the interrogator, is given the task of trying to determine which player – A or B – is a computer and which is a human. The interrogator is limited to using the responses to written questions to make the determination.

Category of AI

Machine learning (ML) and deep learning (DL) are important branches and core technologies of AI.

- Feature of Deep learning:
  - A science devoted to making machines think and act like humans.
  - Focuses on enabling computers to perform tasks without explicit programming.
  - A subset of machine learning based on artificial neural networks.

ML is a method to achieve AI.
DL is a technique for implementing ML.



Machine Learning

Agenda

- What is Machine Learning?
- What are the two main categories of Machine Learning?
- What are some examples of Machine Learning?
- How does Machine Learning "work"?

What is Machine Learning?

*One definition: "Machine Learning is the semi-automated extraction of knowledge from **data**"*

Data-Driven. Learn from data and self-optimize.

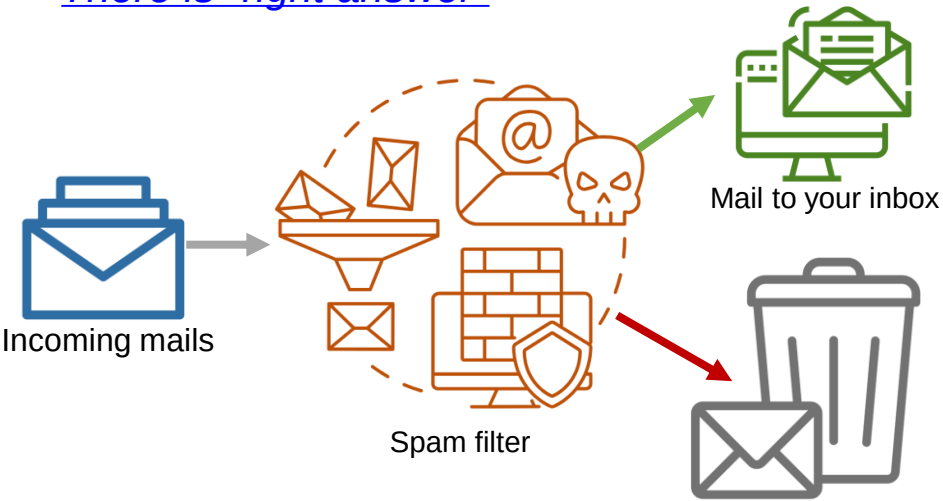
Good interpretability, intuitive and easy to understand, and able to clearly demonstrate the impact of features on predictions.

What are the two main categories of Machine Learning?

- Supervised learning:

Making predictions using data

Example: Is a given email "spam" or "ham"?
There is *an outcome* we are trying to predict
There is "right answer"

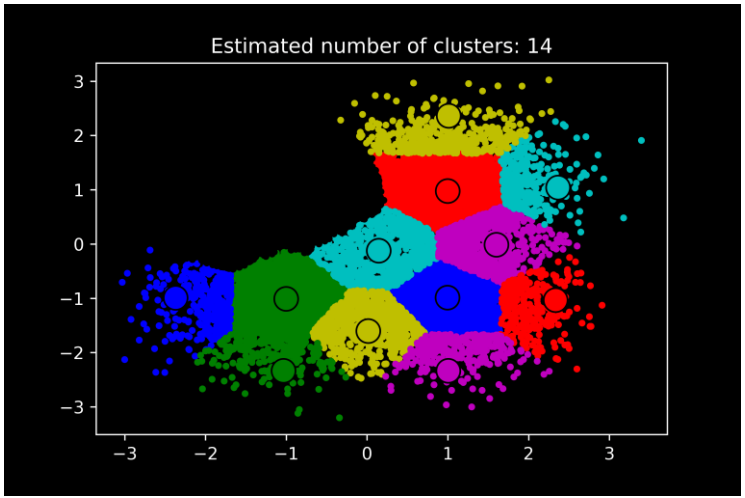


AI helps us to filter spam email daily!

- Unsupervised learning:

Extracting structure from data

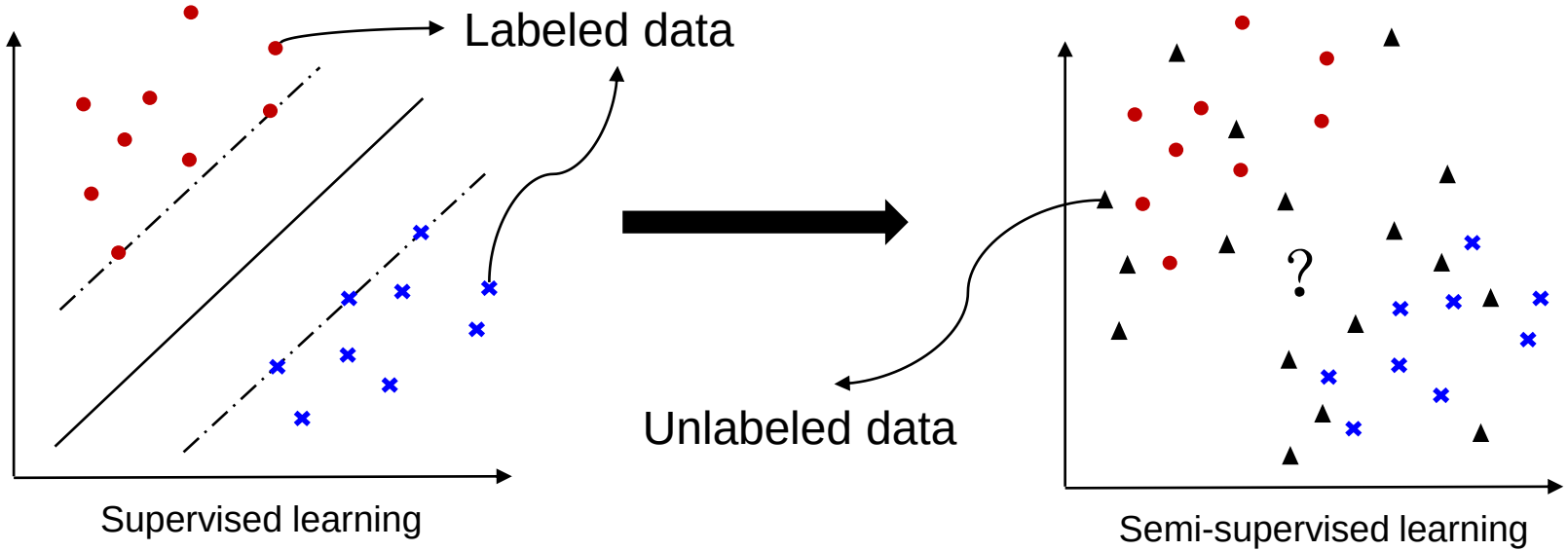
Example: Segment grocery store shoppers into clusters that exhibit similar behaviors
There is no "right answer"



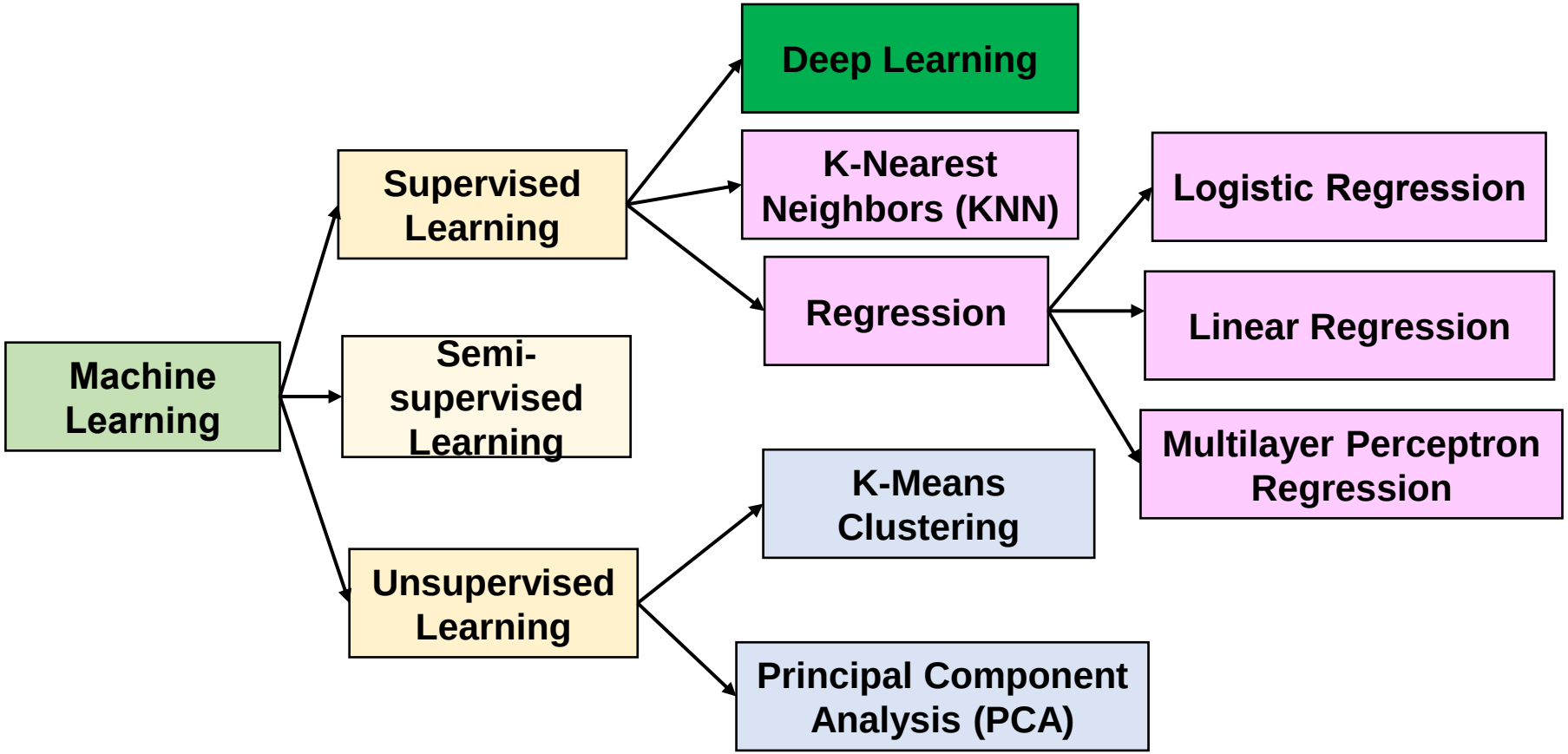
[Source]

Semi-supervised Learning

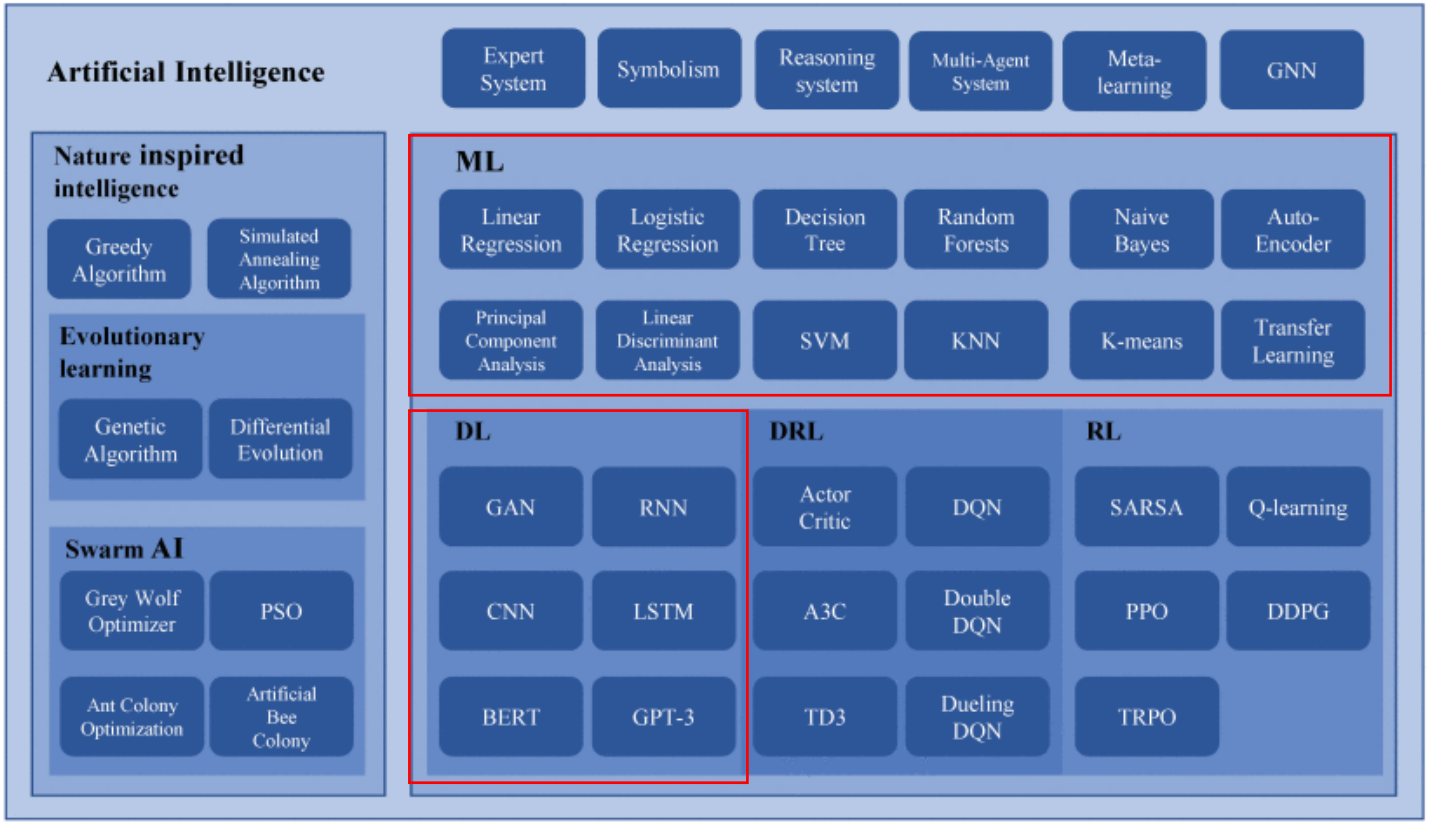
As the name suggests, is the task of producing a prediction rule given example data predictions (labeled data) and extra data without any prediction labels (unlabeled data).



Framework of the main categories



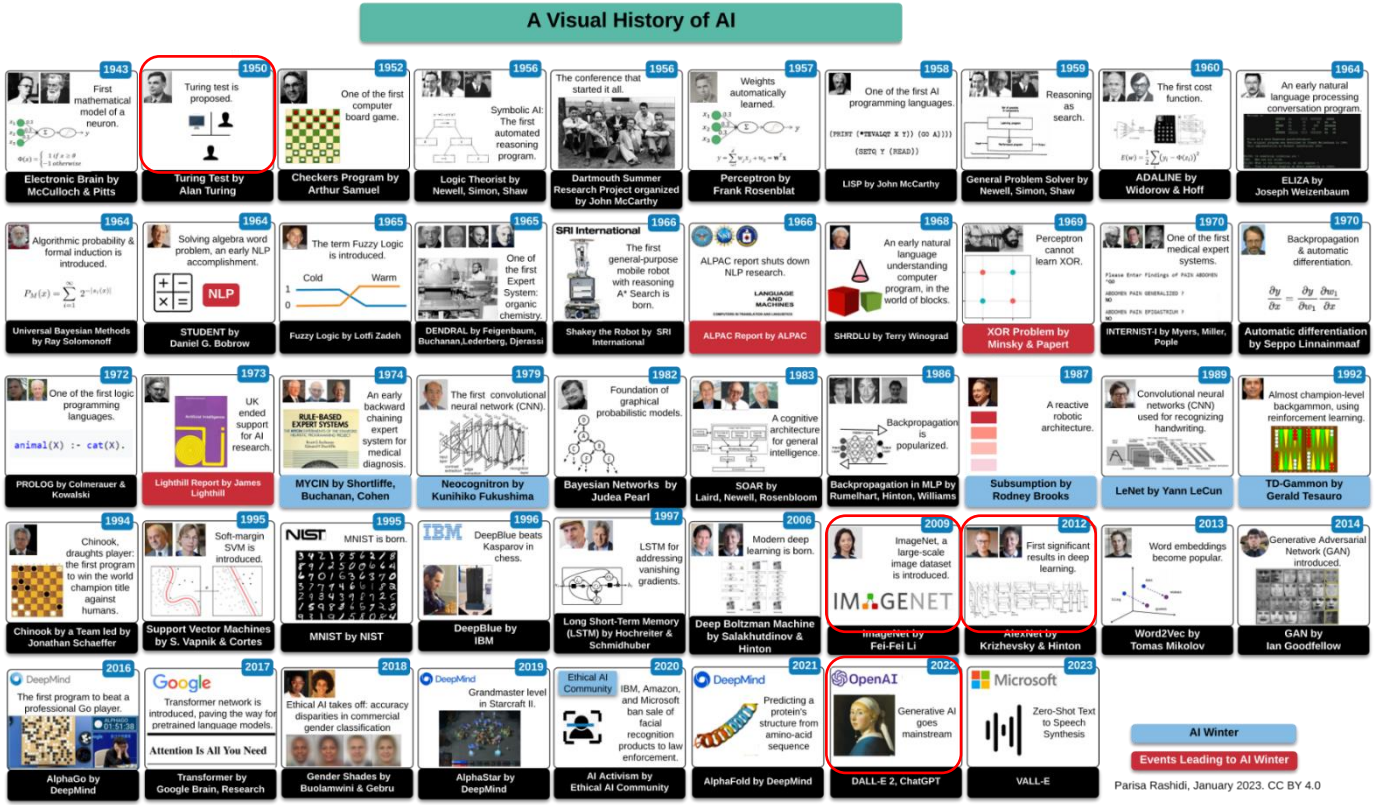
Classification



https://ieeexplore.ieee.org/document/10113154

Department of Aeronautical and Aviation Engineering 航空及民航工程學系
THE HONG KONG POLYTECHNIC UNIVERSITY 香港理工大學

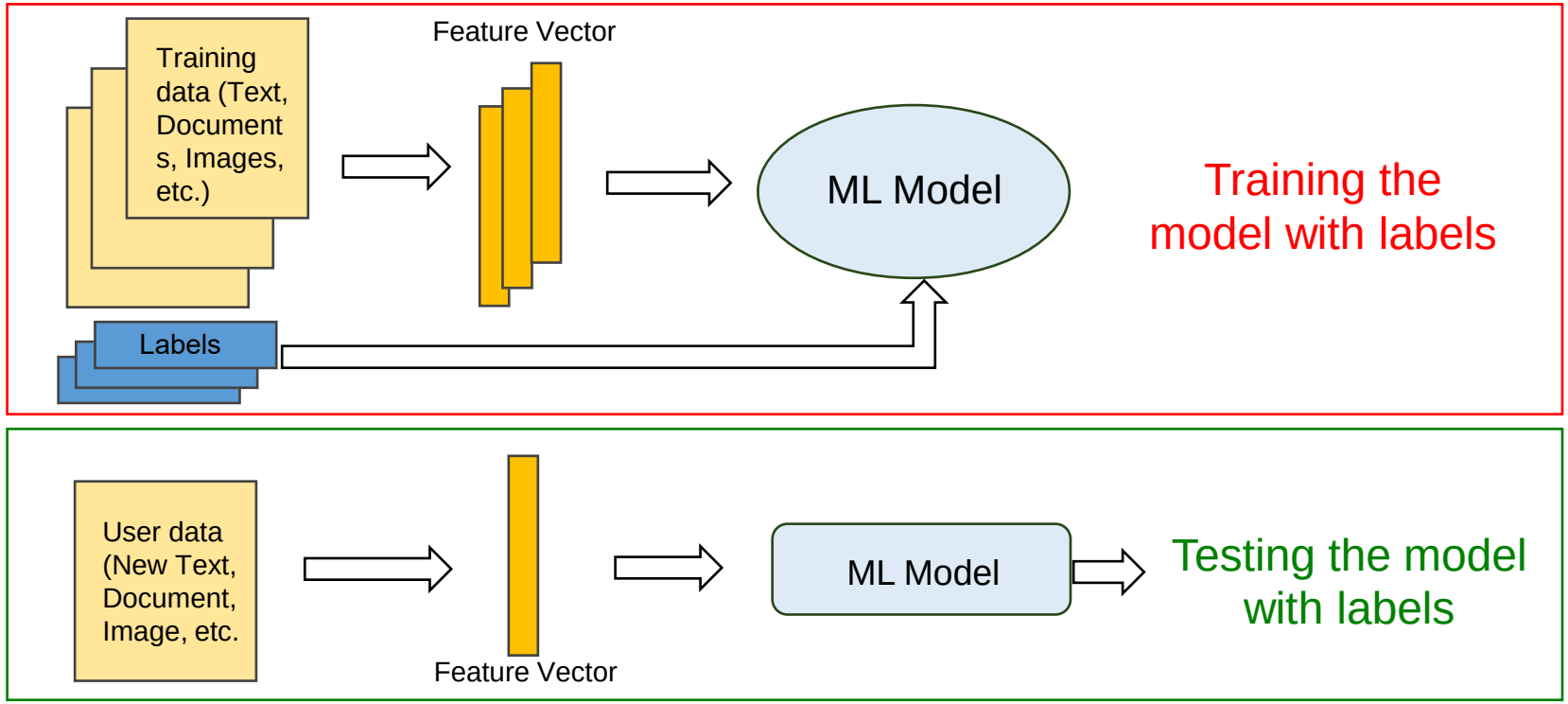
AI Development Journey



https://figshare.com/articles/figure/AI_History_svg/12363890

How does Machine Learning "work"?

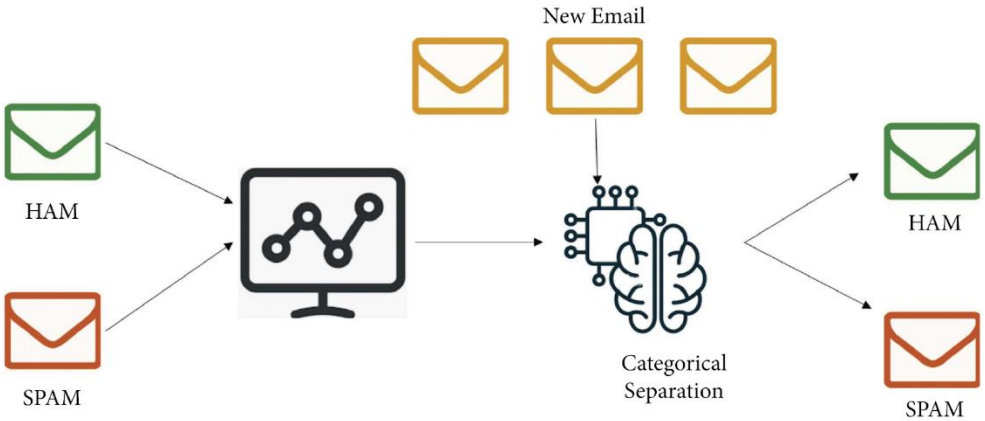
The primary goal of supervised learning is to build a model that "generalizes": It accurately predicts the future rather than the past!

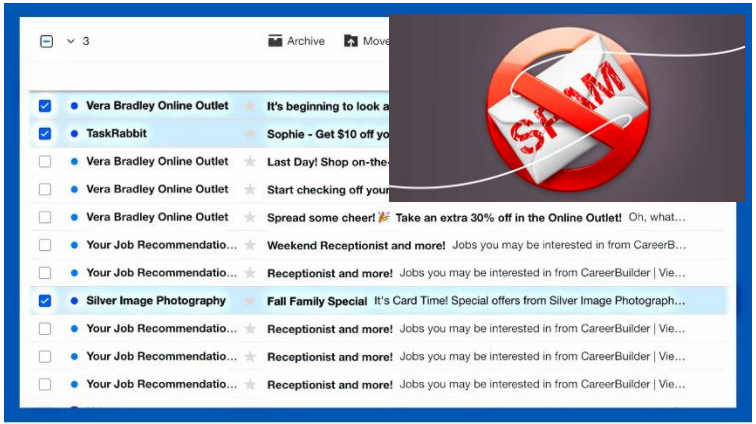


What are the key features of the spam email?

Think about this

Machine learning for Classification of “SPAM emails”





- HAM and SPAM emails are labelled manually to train the AI model (AI network).
- The AI model is used to classify the HAM and SPAM emails daily.
- The power of the AI model with more data!

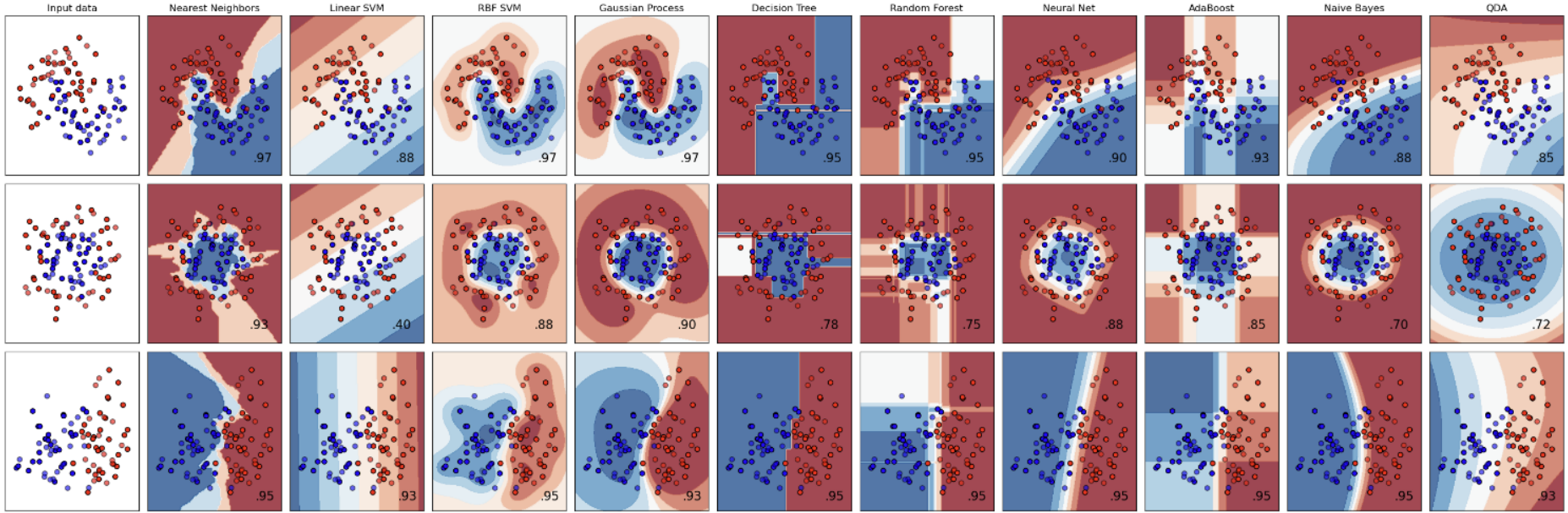
```
Random Forest Accuracy: 0.98
confusion Matrix : [[888   1]
 [ 24 121]]
Precision Score:  0.9918032786885246
```

The Random Forest model achieved an accuracy of 98%, indicating its ability to accurately classify emails as spam or ham. In the confusion matrix, out of 1015 emails, 888 were correctly classified as ham, while only 1 was incorrectly classified. For spam emails, 121 were correctly identified, with 24 being misclassified. The Precision Score, measuring the proportion of correctly classified spam emails among all emails classified as spam, was 99.18%.

The accuracy reach 0.992 using machine learning!

Quick look on the existing classifiers

names = ["Nearest Neighbors", "Linear SVM", "RBF SVM", "Gaussian Process", "Decision Tree", "Random Forest", "Neural Net", "AdaBoost", "Naive Bayes", "QDA",]

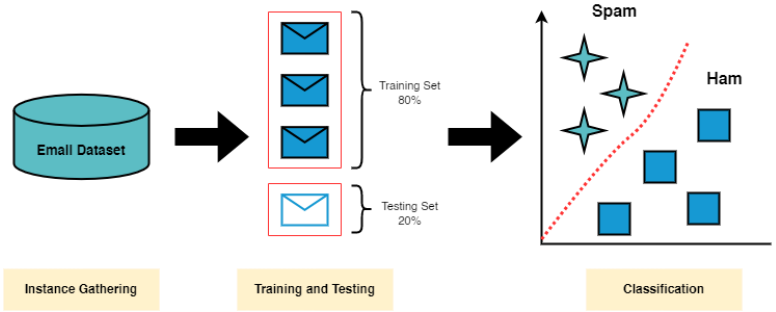


The plots show training points in solid colors and testing points semi-transparent.
The lower right shows the classification accuracy on the test set.

Example: Regression for classification of “SPAM emails”

Regression analysis: identifying quantitative relationships between two or more variables that are interdependent.

$$y = f(x_1, x_2 \dots x_n)$$



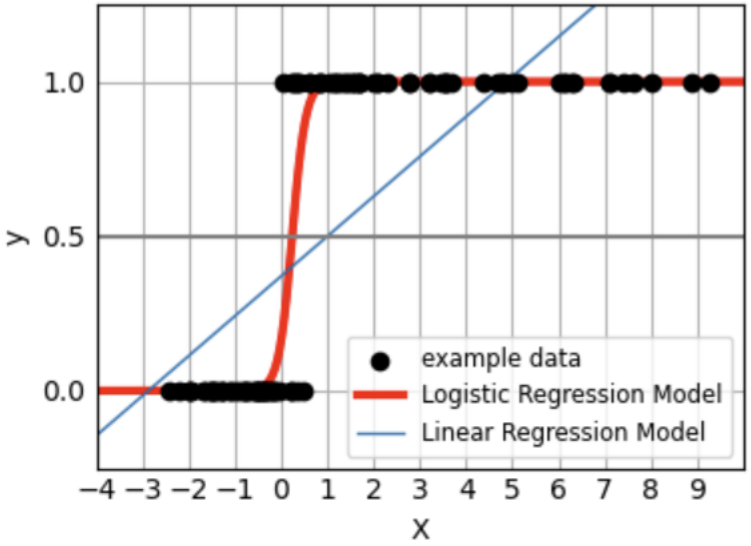
Regression

- **Number of variables**
  - **one-dimensional regression:** $y = f(x)$
  - **multivariate regression :** $y = f(x_1, x_2 \dots x_n)$
- **Functional Relationships**
  - **linear regression:** $y = ax + b$
  - **Non-linear regression:** $y = ax^2 + bx + c$

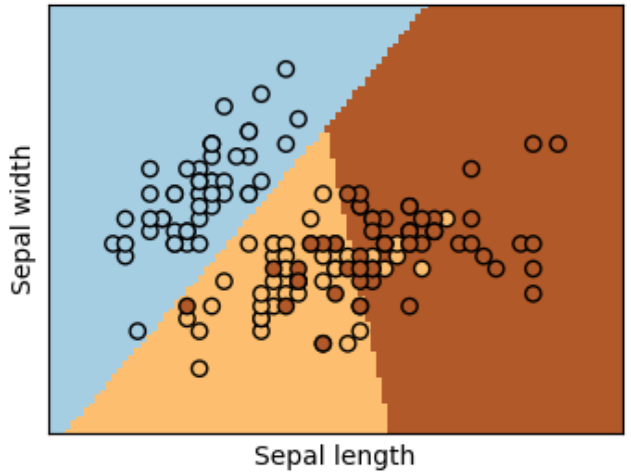
Logistic regression VS Linear regression

Logistic regression *estimates the probability* of an event occurring, such as voted or didn't vote, based on a given data set of independent variables.

$$\text{logistic}(pi) = \frac{1}{(1+e^{-pi})}$$



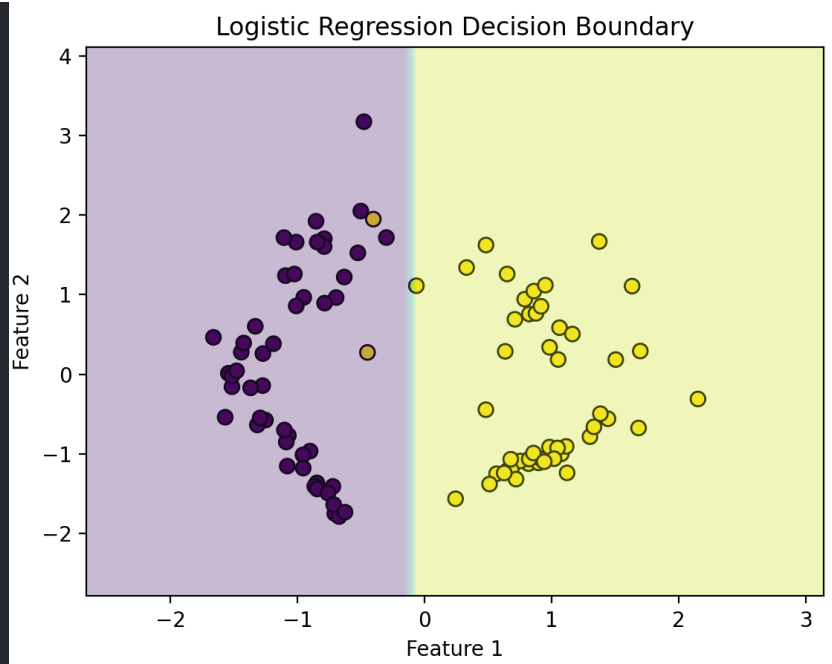
Logistic Regression 3-class Classifier



[Source]

Try the regression in one second!

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.datasets import make_classification

# Generate a synthetic dataset
X, y = make_classification(n_samples=100, n_features=2, n_informative=2, n_redundant=0, random_state=1)

# Create a logistic regression model
model = LogisticRegression()

# Fit the model to the data
model.fit(X, y)

# Plot the data points
plt.scatter(X[:, 0], X[:, 1], c=y, cmap='viridis', edgecolor='k', s=50)

# Create a mesh to plot the decision boundary
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, 0.1),
                     np.arange(y_min, y_max, 0.1))

# Predict the function value for the whole grid
Z = model.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)

# Plot the decision boundary
plt.contourf(xx, yy, Z, alpha=0.3, cmap='viridis')

# Add labels and a title
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.title('Logistic Regression Decision Boundary')

# Display the plot
plt.show()
```

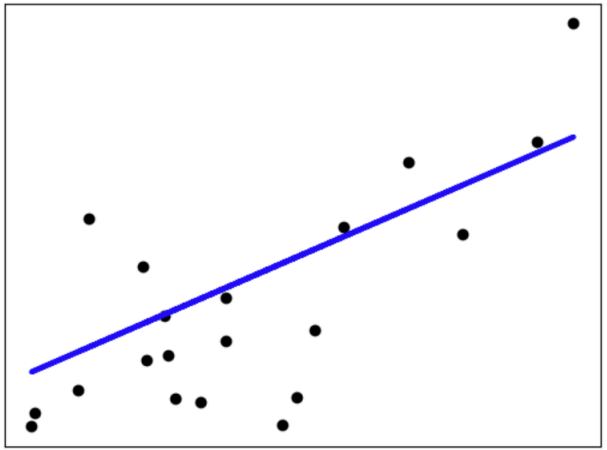


[Ask ChatGPT with:]

generate the python code for plot the logic regression function

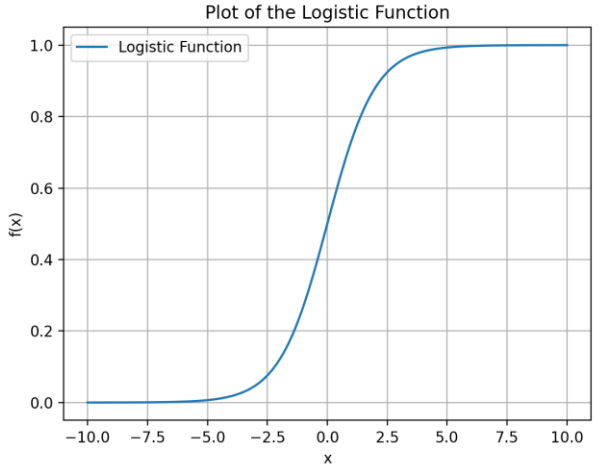
Logistic regression VS Linear regression

Linear regression attempts to draw a straight line that will best minimize the residual sum of squares between the observed responses in the dataset, and the responses predicted by the linear approximation.



[Source]

Logistic regression



Logistic regression VS Linear regression

- Each type of *linear regression*, it attempts to *draw a line that best fits* through a set of data points, which is usually calculated using the least squares method.

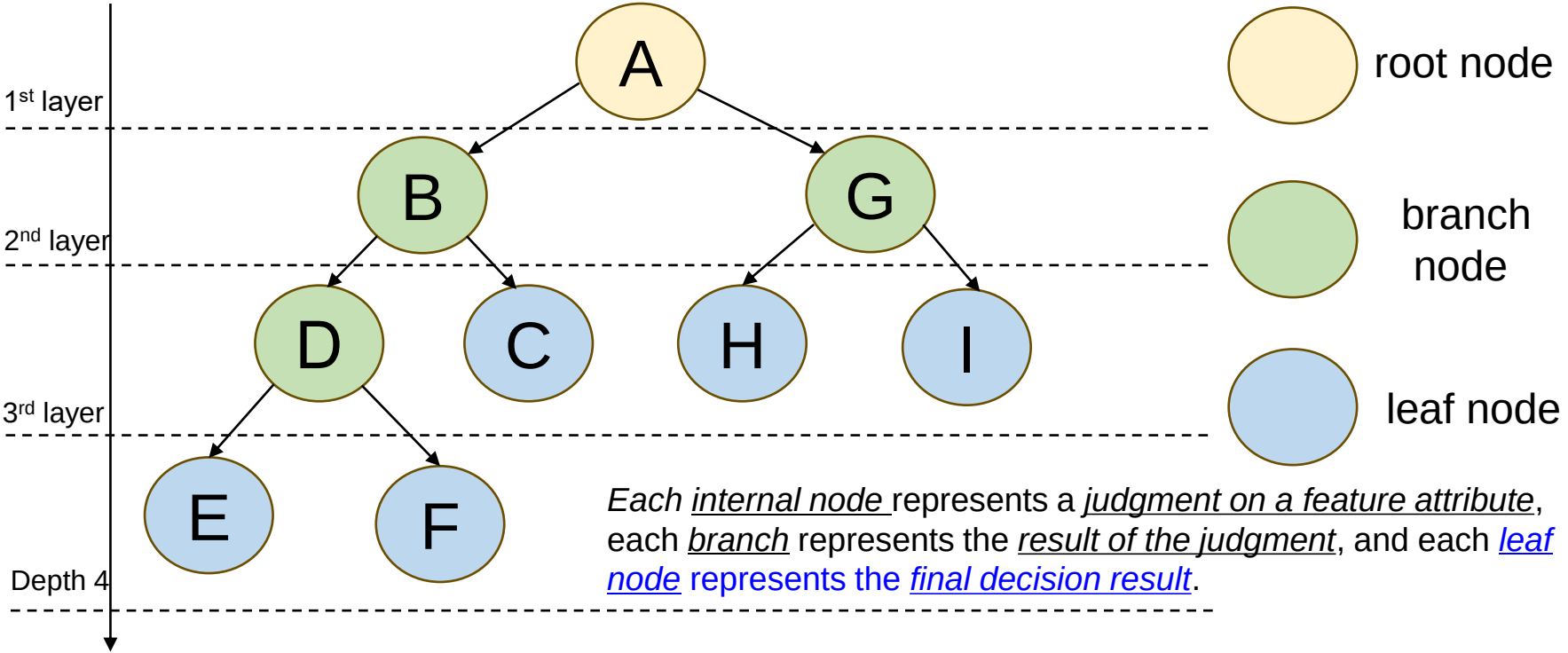
- Logistic regression is used to make predictions about categorical variables versus continuous variables. Categorical variables can be true or false, yes or no, 1 or 0, and so on. The unit of measure also differs from linear regression in that it *produces probabilities*, but the logit function converts an S-curve into a straight line.

- *Linear regression is usually easier to understand*. Linear regression also does not require as large a sample size as Logistic regression requires enough samples to represent the values in all response categories.

- Without a larger representative sample, the model may not have enough statistical power to detect significant effects.

Decision Tree

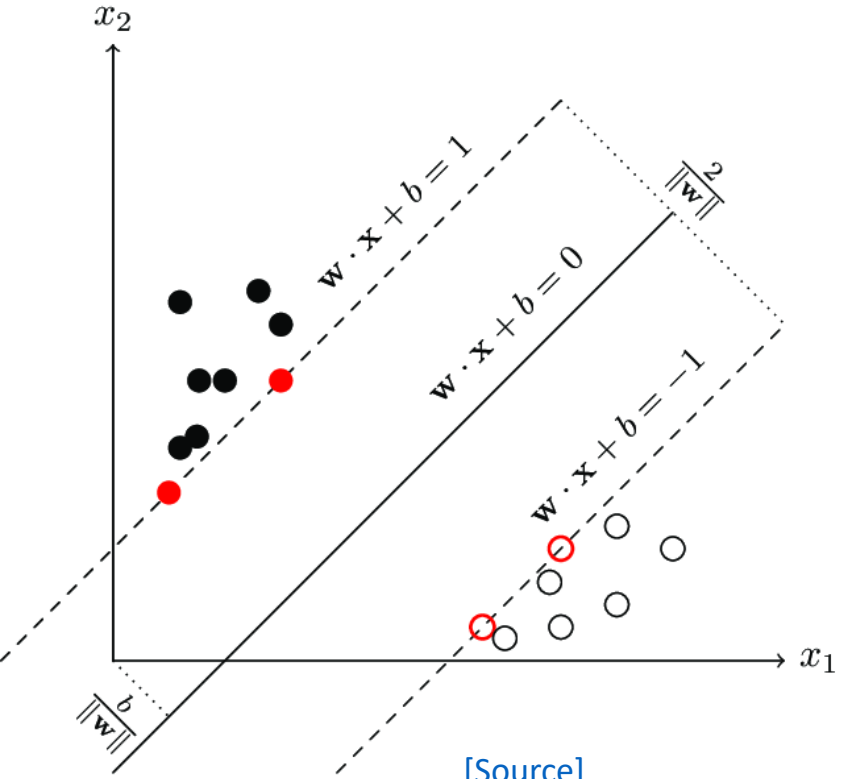
The goal is to create a model that predicts the value of a target variable *by learning simple decision rules* inferred from the data features.



Each internal node represents a judgment on a feature attribute, each branch represents the result of the judgment, and each leaf node represents the final decision result.

Support Vector Machines (SVM)

A classifier defined to maximize intervals over the feature space.



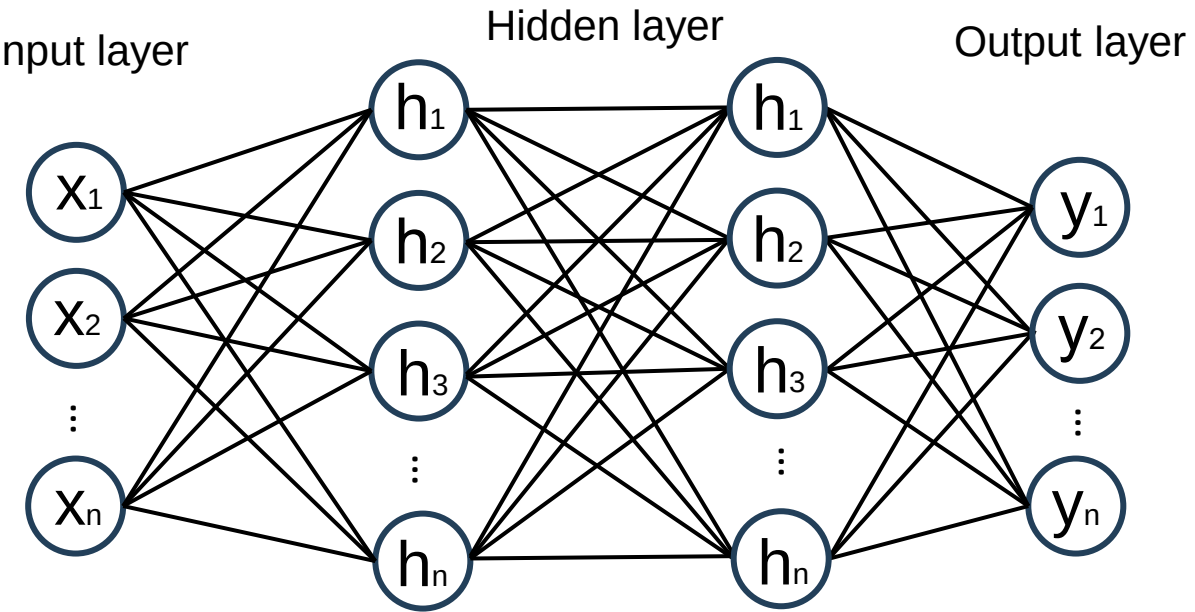
[Source]

$w \cdot x + b = 0$ is the *separating hyperplane*, and for a linearly divisible dataset, there are infinitely many such hyperplanes (i.e., perceptual machines). *The variable w is the normal vector of the* hyperplanes.

"support vectors" are the data points that are closest to the hyperplane. These data points are used to determine the location and orientation of the hyperplane.

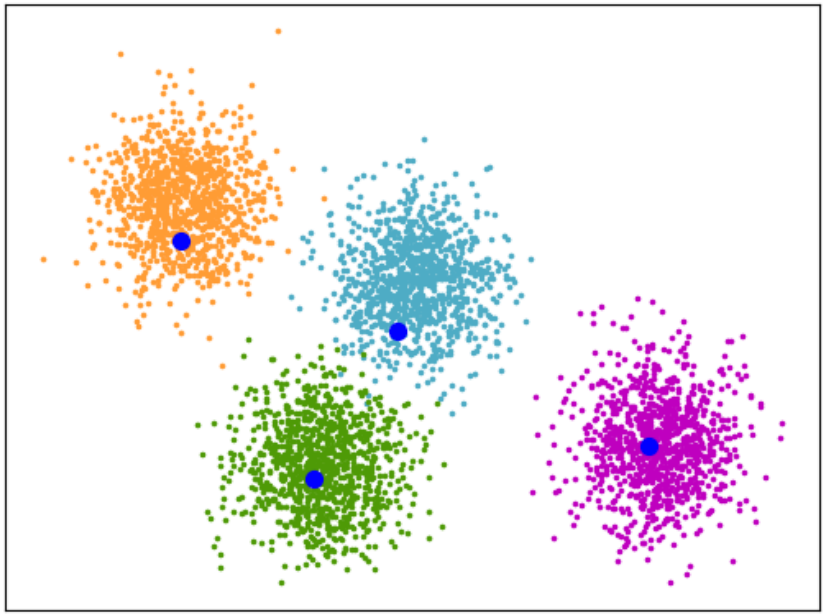
Multilayer perceptron (MLP)

A multilayer perceptron (MLP) is a name for a modern feedforward artificial neural network, consisting of fully connected neurons with a nonlinear activation function, organized in at least 3 layers, notable for being able to distinguish data that is not linearly separable.



Clustering

Clustering algorithm is an ***unsupervised learning*** method for partitioning a data set into different classes or clusters according to a particular criterion (e.g., distance), *so that data objects within the same cluster are as similar as possible, while data objects not in the same cluster are as different as possible.*

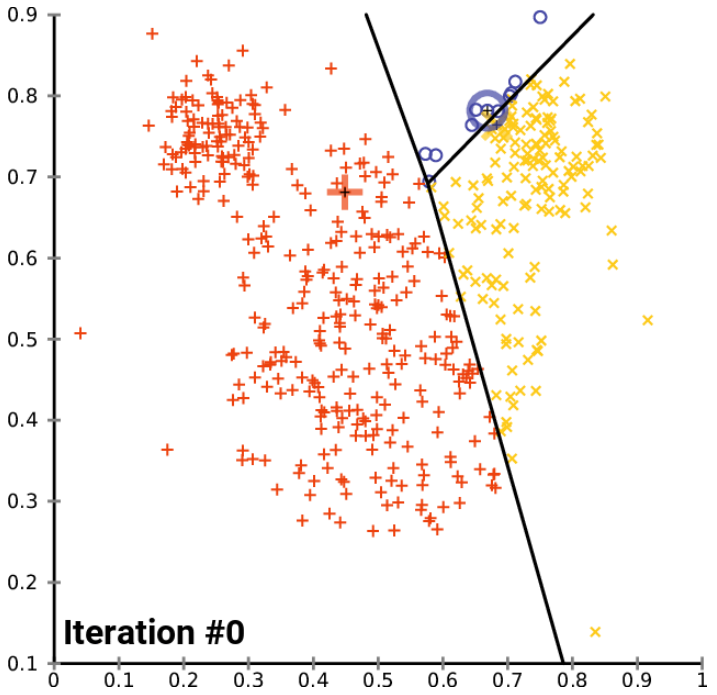


[Source]

K-means

It is based on centroid-based clustering.

The basic principle : through an *iterative process*, the data points are divided into the class clusters represented by the closest class cluster centroids, and then the centroids of the class clusters are recalculated (averaged) based on all the points within each class cluster.



[Source: wikipedia]

K-means

- **Choice of number of clusters (K):**

  The first step in the K-Means algorithm is to determine *how many clusters* to divide the data into.

- **Distance measure:**

  K-Means uses Euclidean distance to measure the similarity between data points, but other distance measures can be chosen according to the specific problem.

- **Center of Mass:**

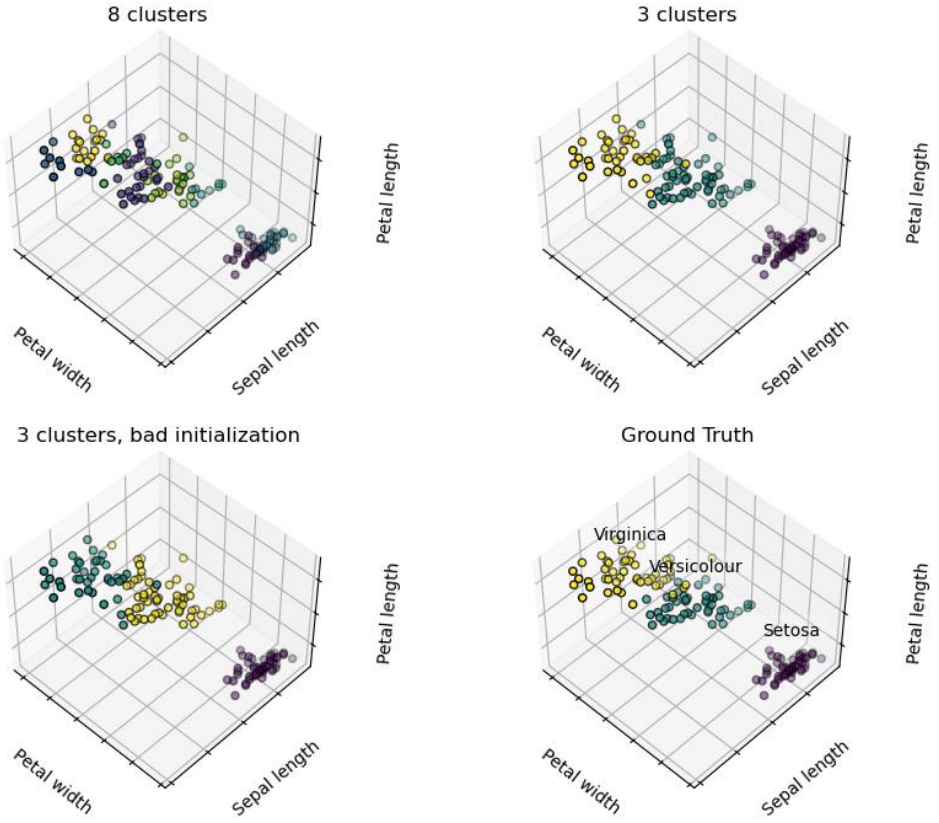
  Each cluster has a center of mass, which is the mean of all data points within that cluster. The center of mass represents the center of the cluster.

- **Optimization objective:**

  The optimization objective of K-Means is to minimize the sum of the distances from each data point to the center of mass of the cluster to which it belongs.

K-means analysis

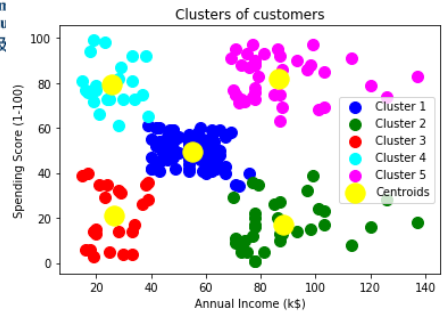
- *top left:* What a K-means algorithm would yield using 8 clusters.
- *top right:* What using three clusters would deliver.
- *bottom left:* What the effect of a bad initialization is on the classification process.
- *bottom right:* The ground truth.

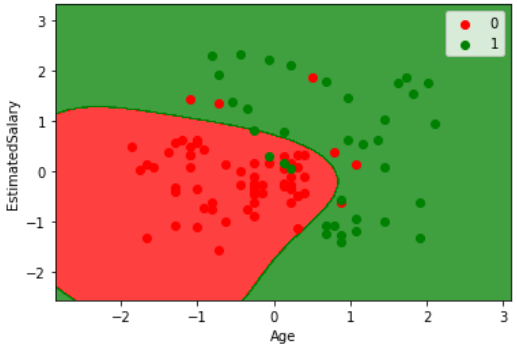


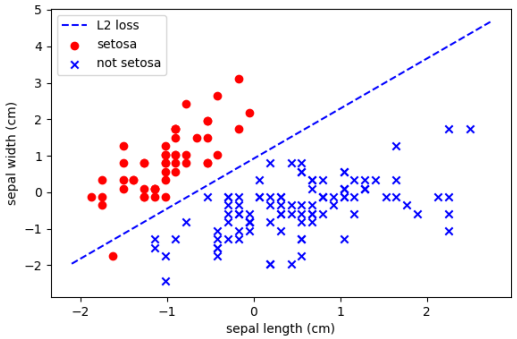
[Source]

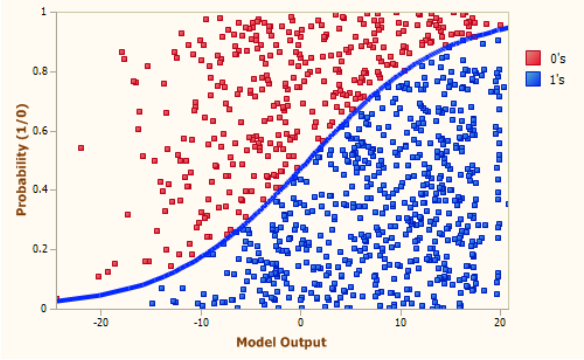
Short summary!

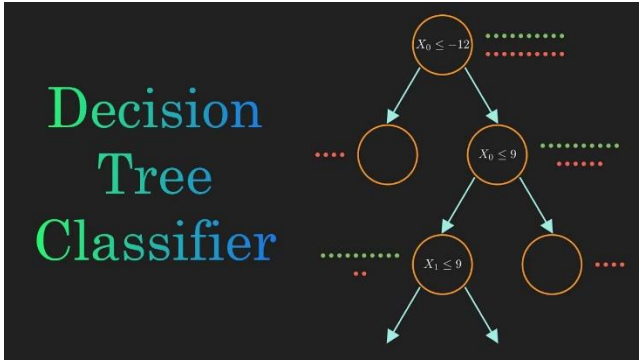
Methods	Features
Linear regression	Suitable for simple linear regression case
Logistic regression	Suitable for non-linear regression
Decision tree	Rule based classification problem
SVM	Complex. High dimension non-linear regression
K-means	Given known number of clusters, without labels









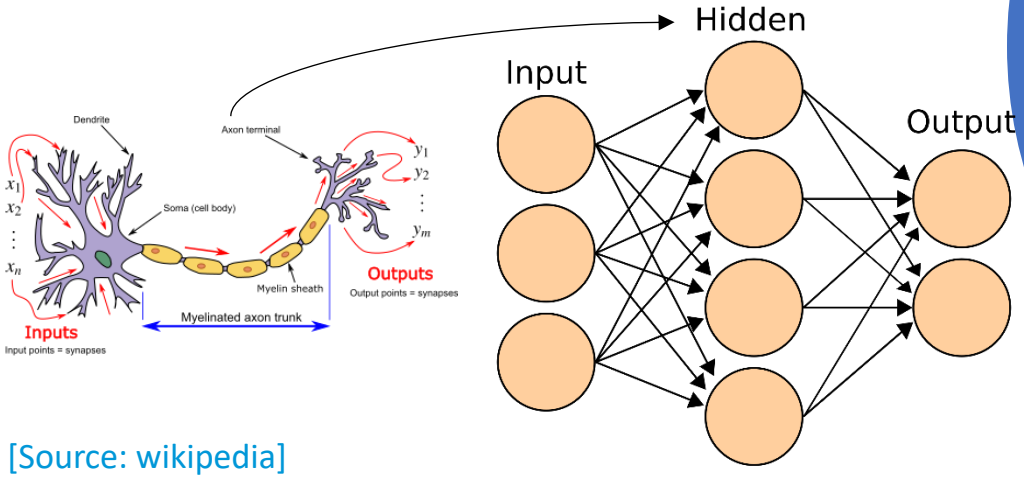


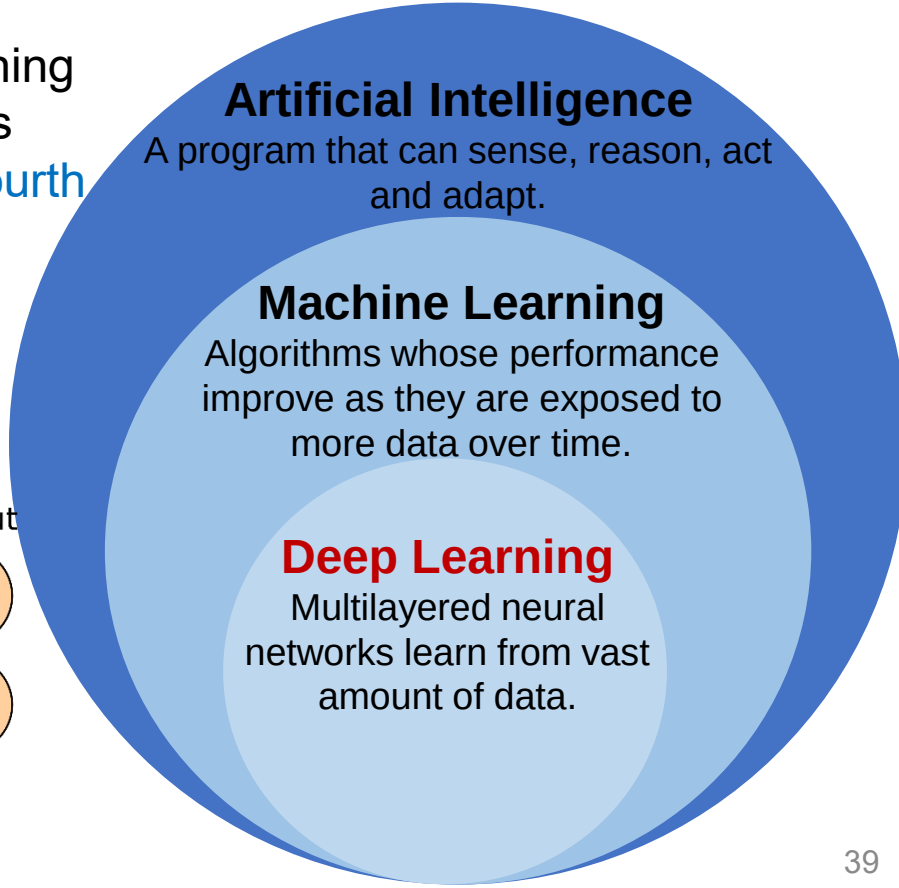
Deep Learning

Deep learning (DL), a branch of machine learning (ML) and artificial intelligence (AI) is nowadays considered as a core technology of today's Fourth Industrial Revolution (4IR or Industry 4.0).

⇩

Artificial Neural Network (ANN)



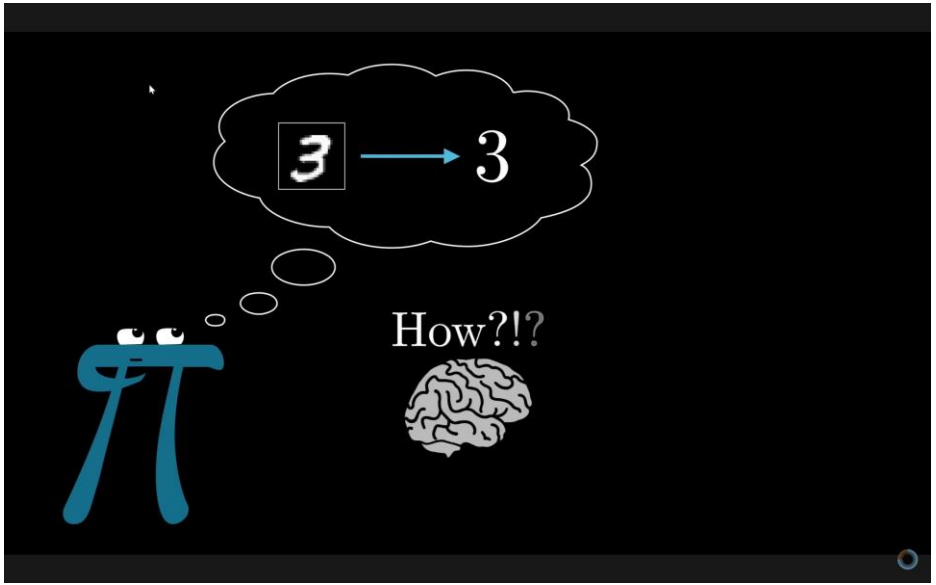


[Source: wikipedia]

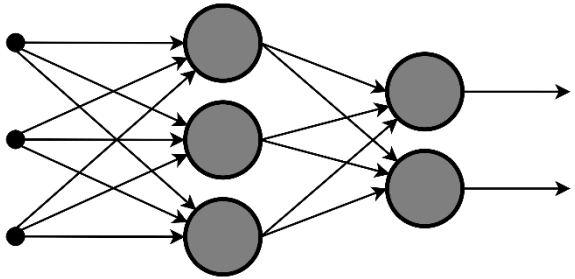
General Idea of Deep Learning

What is a neural network?

https://youtu.be/aircAruvnKk?si=KDyY6OidrVVxtKir

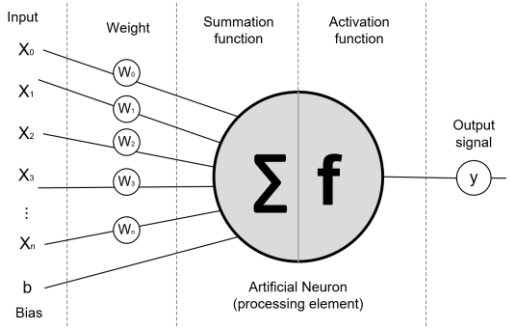


Structure



Learning

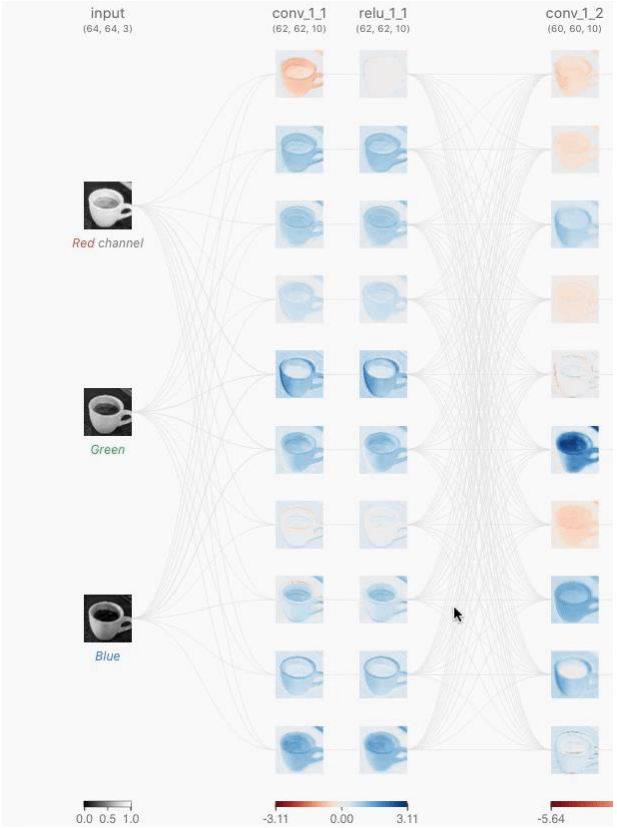
The process of building a model by correcting the weights of each layer.



$$y^k = f(W^k * x + b^k)$$

What is a Convolutional Neural Network?

- A **tensor** can be thought of as an **n-dimensional matrix.**
- A **neuron** can be thought of as a function that takes in multiple inputs and yields a single output.
- A **layer** is simply a collection of neurons with the same operation, including the same hyperparameters.
- **Kernel weights and biases**, while unique to each neuron, are tuned during the training phase, and allow the classifier to adapt to the problem and dataset provided.
- A CNN conveys a **differentiable score function**, which is represented as **class scores** in the visualization on the output layer.



https://poloclub.github.io/cnn-explainer/

Conventional machine learning VS deep learning

1. **Feature Engineering**:
    1. **Machine Learning**: Traditional machine learning methods often require manual feature engineering. This means that domain experts need to identify and extract relevant features from the email data, such as specific keywords, frequency of certain terms, or metadata.
    2. **Deep Learning**: Deep learning models, particularly neural networks, can automatically learn features from raw data, reducing the need for manual feature engineering. This can be advantageous in capturing complex patterns in the data.
2. **Complexity and Flexibility**:
    1. **Machine Learning**: Models like decision trees, support vector machines, or logistic regression are generally simpler and may not capture complex patterns as effectively as deep learning models.
    2. **Deep Learning**: These models can capture more complex, non-linear relationships in the data, which might be beneficial for detecting sophisticated spam tactics.
3. **Adaptability**:
    1. **Machine Learning**: May require more frequent updates and manual tuning to adapt to new spam tactics.
    2. **Deep Learning**: Can potentially adapt better to new patterns without explicit reprogramming, given enough data.

RNN

Recurrent Neural Networks are mainly used to process sequential data, and are able to capture the temporal dependencies in the data.

➢ Advantages

- Processing Sequential Data. RNNs are good at processing **sequential data with temporal relationships**, such as **text**, **speech**, etc.
- Parameter Sharing. **Sharing parameters at the time step** reduces the number of parameters.

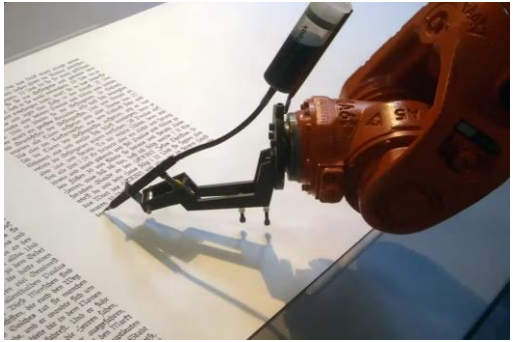
➢ Disadvantages

- **Long** Dependency Problem. Difficult to capture temporal dependencies over long distances, prone to **gradient vanishing** or **gradient explosion problems.**
- Low computational efficiency. **Difficult to parallelize** the computation.

Application of RNNs

- **Machine Translation**

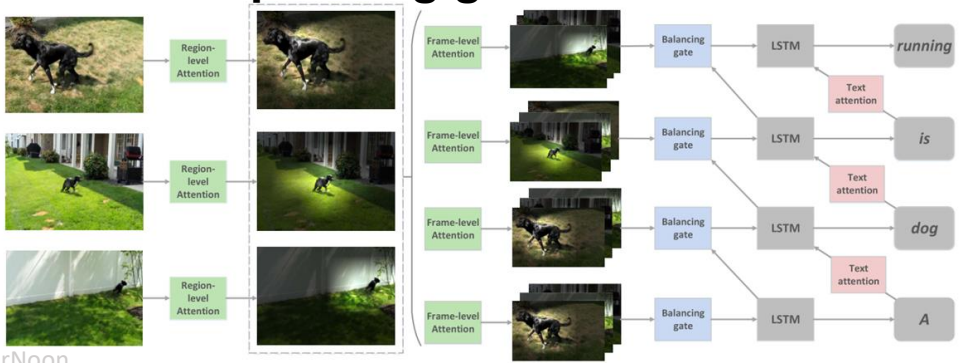
- **Text Generation**



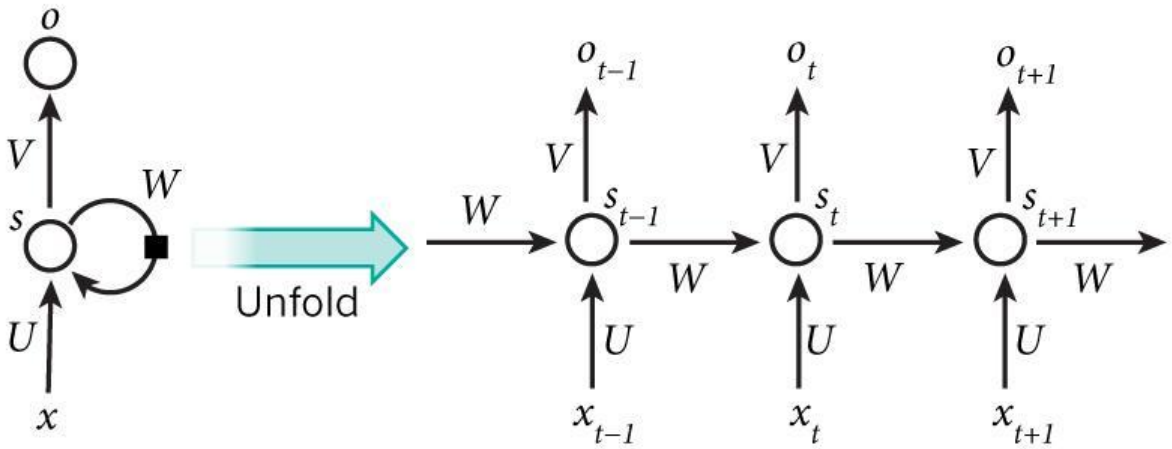
- **Speech Recognition**

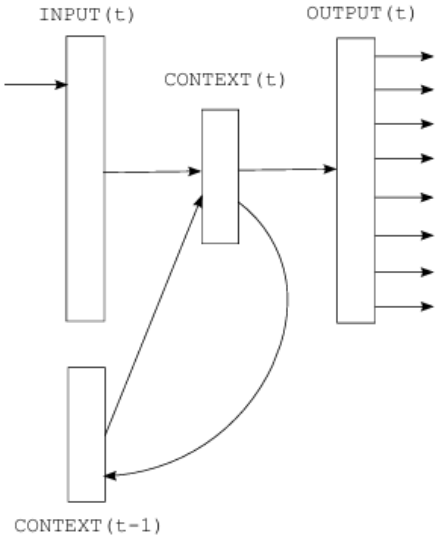
- **Time-Series Forecasting**

- **Video captioning generation**



Memory capabilities





RNNs have received the most success when working with sequences of words and paragraphs, generally in the field of **natural language processing(NLP).**

https://www.semanticscholar.org/paper/Recurrent-neural-network-based-language-model-Mikolov-Karafi%C3%A1t/9819b600a828a57e1cde047bbe710d3446b30da5

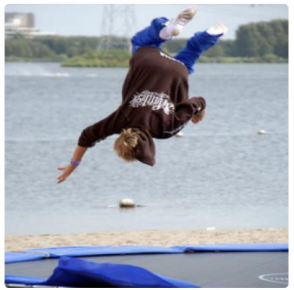
Application of RNNs in Text Generation

RNN generate sentence descriptions from images.

"man in black shirt is playing guitar."

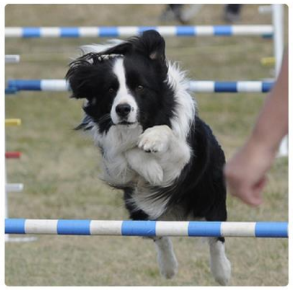
"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."



"boy is doing backflip on wakeboard."

"girl in pink dress is jumping in air."



"black and white dog jumps over bar."

"young girl in pink shirt is swinging on swing."

"man in blue wetsuit is surfing on wave."

Deep Visual-Semantic Alignments for Generating Image Descriptions (stanford.edu)

GAN

Generative Adversarial Networks as the name suggests, consist of a **generative network** and a network that fights against the generative network, i.e., a **discriminative network.**

The training optimization objective of GAN:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

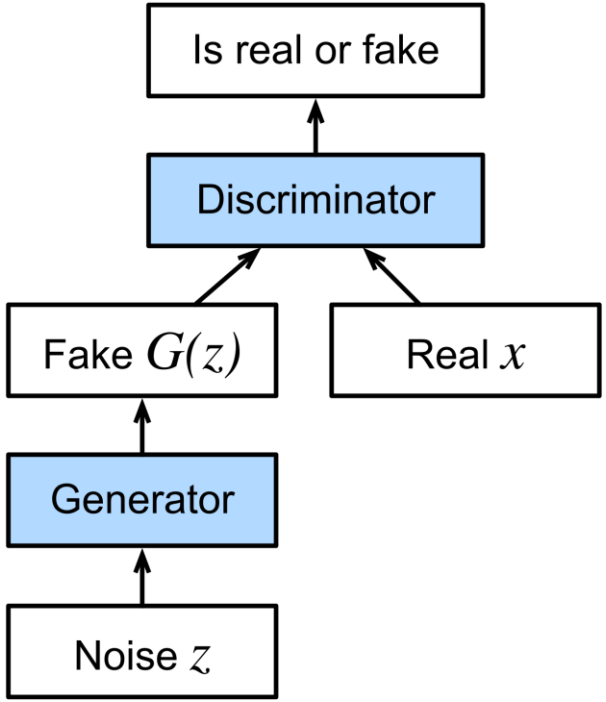
- V(D,G) represents degree of difference between the generated samples and the real samples, which can use the cross-entropy loss of binary categorization (two categories: true and false).
- The discriminator D are updated by maximizing the cross-entropy loss V(D,G) with the generator fixed.
- Indicates that the **generator is to minimize this cross-entropy loss V(D,G)** in case the **discriminator maximizes the cross-entropy loss V(D,G) for true and false pictures.**

https://arxiv.org/abs/1406.2661

Department of Aeronautical and Aviation Engineering 航空及民航工程學系
THE HONG KONG POLYTECHNIC UNIVERSITY 香港理工大學

Architecture of GAN

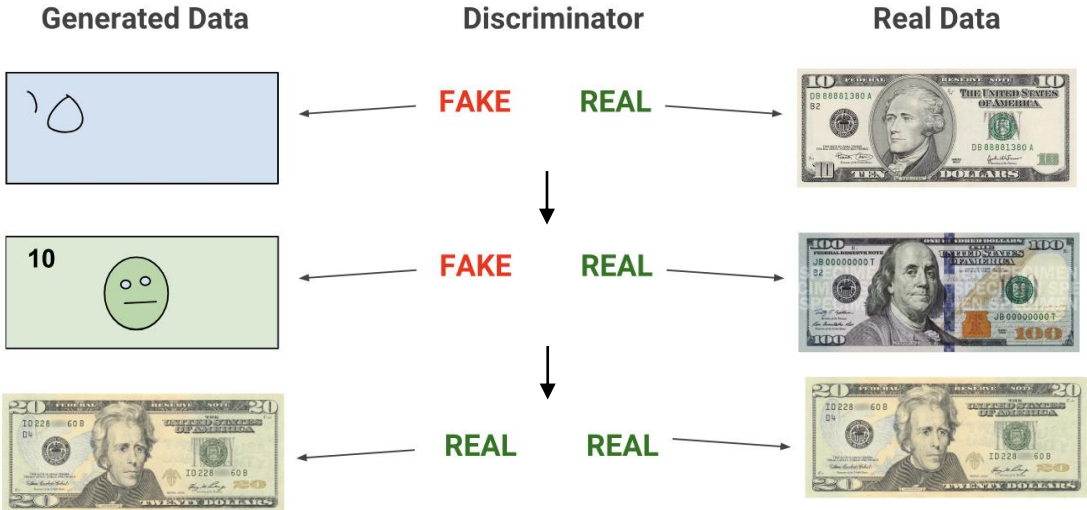
Overview of GAN Structure



- The **generator** learns to generate plausible data. The generated instances become negative training examples for the discriminator.

- The **discriminator** learns to distinguish the generator's fake data from real data. The discriminator penalizes the generator for producing implausible results.

- *When training begins,*

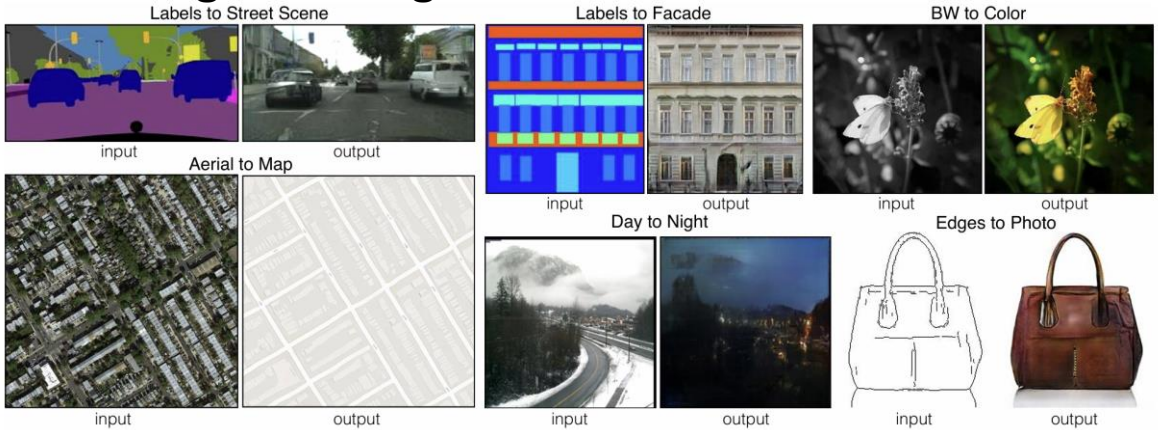


Generative adversarial network – Wikipedia
Overview of GAN Structure | Machine Learning | Google for Developers

Application of GANs

➢ Generate Images

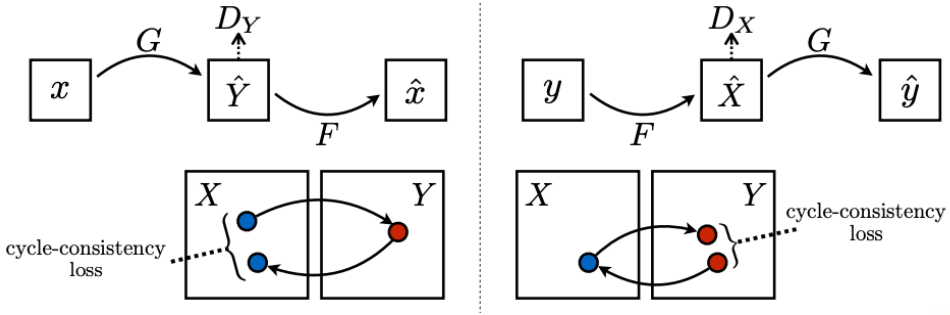
➢ Image-to-Image Translation

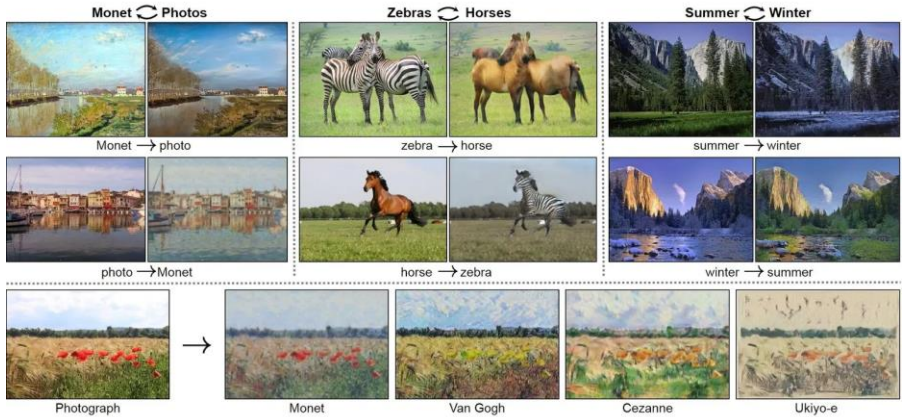


➢ Text-to-Image Translation (text2image)

CycleGAN

*CycleGAN can **transfer** the properties of both images to each other.*

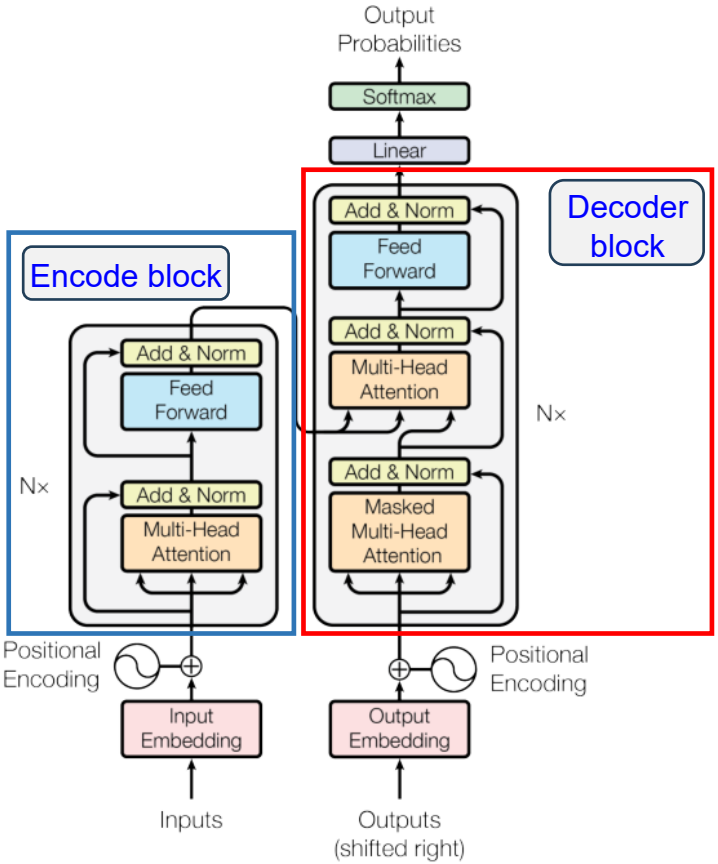




CycleGAN typically uses two generators and discriminators. The idea behind it is that image A is fed into the generator, and it generates a certain image G(A). The same image G(A) is fed into another generator to reconstruct the original image F(A).

Generative Adversarial Networks and Some of GAN Applications: Everything You Need to Know (neptune.ai)

Transformer, Attention is all you need.



Encoder

- Process the input sequence and convert it into a representation that captures the meaning and context.

Decoder

- Receive the encoder's output and generate the desired output sequence.

Attention Mechanism

- Focus on specific parts of the input sequence, improving its understanding.

V. Ashishet, al, "Attention is All you Need", Advances in Neural Information Processing Systems, 2017.

In class task: Example of Deep Learning for Drones

VisDrone Dataset

- *The dataset is wide-area aerial images taken by UAVs at different locations and altitudes.*
- *The training set includes 6471 images and the validation set includes 548 images with a resolution of about 2000*1500 pixels.*
- *Including 10 categories (pedestrian, person, car, van, bus, truck, motor, bicycle, awning-tricycle, and tricycle).*

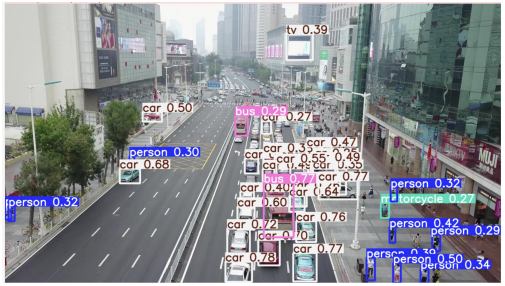
Run the Yolo detection with Colab

Link to Colab: https://colab.research.google.com/

What is Colab?: Colab is a hosted Jupyter Notebook service that requires no setup to use and provides free access to computing resources, including GPUs and TPUs. Colab is especially well suited to machine learning, data science, and education. Open Colab New Notebook.

```
from IPython import get_ipython
from IPython.display import display
# %%
!pip install torch torchvision torchaudio
!pip install yolov5
# %%
import torch

model = torch.hub.load('ultralytics/yolov5', 'yolov5s', pretrained=True)
from PIL import Image
import matplotlib.pyplot as plt

# Replace with the correct path to your image within the Colab environment
img_path = '/content/0000001_04527_d_0000008.jpg'
img = Image.open(img_path)
plt.imshow(img)
plt.axis('off')
plt.show()
results = model(img)

results.show()
results.save('/content/pre_0000001_04527_d_0000008.jpg')
```

Find the source code at: https://github.com/weisongwen/AAE4011-S22425/issues/2



YOLOv5 for Drone Detection (need details)

Drone target detection starts quickly!!! In jupyter notebook

```
!pip install torch torchvision torchaudio
!pip install yolov5
```

```
import torch
from PIL import Image
import matplotlib.pyplot as plt
```

→ Import necessary package

Import pre-trained models

```
model = torch.hub.load('ultralytics/yolov5', 'yolov5s',pretrained=True)

img_path = "C:/Users/ROG/Desktop/UAS Lecture/VisDrone2019-DET-
val/images/0000001_04527_d_0000008.jpg"  # Drone aerial image
img = Image.open(img_path)
plt.imshow(img)
plt.axis('off')
```

→ Input image

YOLOv5 for Drone Detection

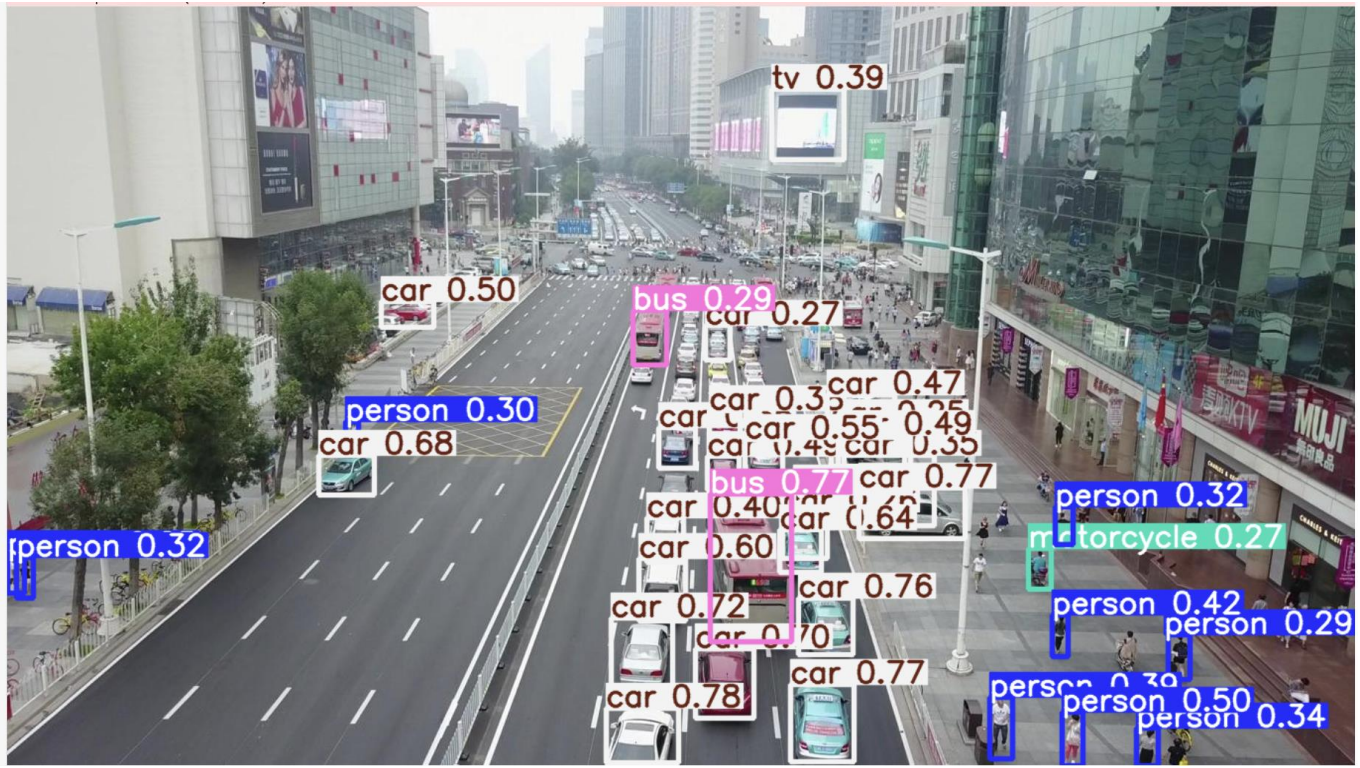
Drone target detection starts quickly!!! In jupyter notebook

```
plt.show()# Reasoning, get detection results
results = model(img)
results.show()# show the results
results.save('output_folder')  # save the detected image, replace it with
the path you want to save it to.
```

Prediction

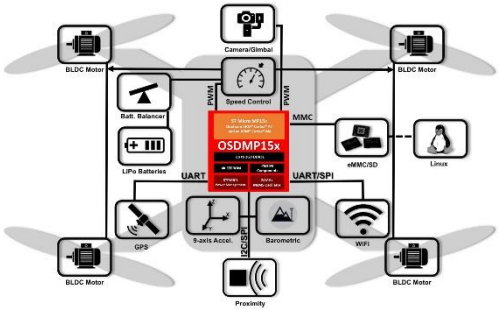
YOLOv5 for Drone Detection

VisDrone2019-DET-val/images/0000295_01600_d_0000029.jpg

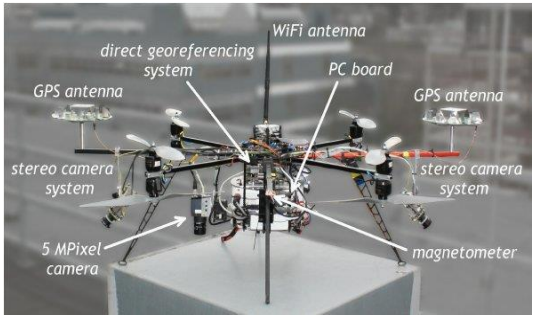


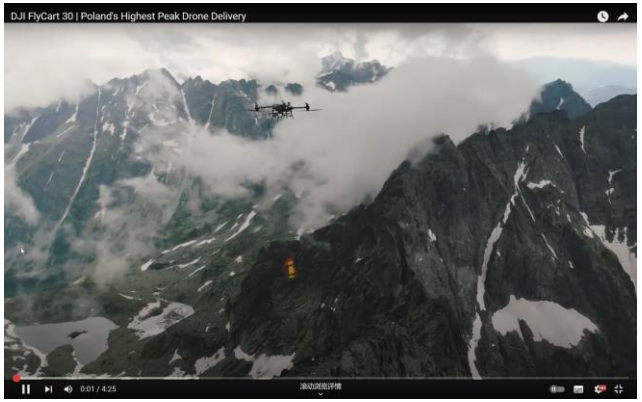
What is Drone?

Components and system for drone



Sensors, perception system for drones



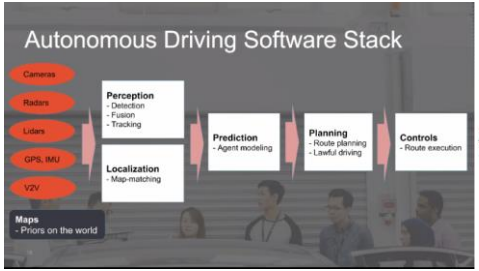


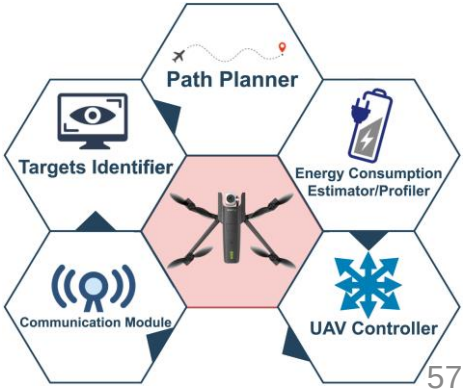
DJI FlyCart 30 | Poland's Highest Peak Drone Delivery

Scene understanding and planning (what is the optimal path to there?)

Perception (Localization and mapping) (where are you?)

System control by wire (how can you to go there?)



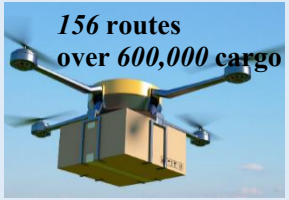


What is it?

The low-altitude economy (LAE) refers to the vertical airspace that extends from 1,000 to 4,000 meters above the ground where civil-manned and unmanned aircraft vehicles operate.

What can it do?

Smart Transportation

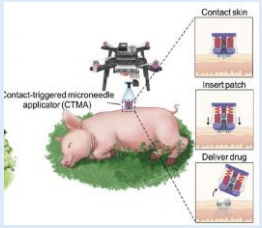


Aerial Delivery [1,2] eVTOL aircraft [3]

Automation

Window cleaning [3] Inspection [4]

Aerial Physic Interaction



Assisting people's work by learning

Medication Injection[5] Collaboration[6]

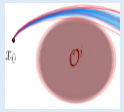
The Interim Regulation on the Administration of the Flight of ***Unmanned Aircraft***, 2024
Regulation of the Shenzhen Special Economic Zone on the Promotion of ***Digital Economy Industries***, 2024


Winnie Tang
Opinion
Flying taxis in Hong Kong? How to kick-start our 'low-altitude economy'
South China Morning Post

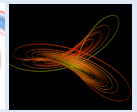
Law and Policy

Dr Winnie Tang is an adjunct professor at the University of Hong Kong, and founder and honorary president of the ***Smart City*** Consortium

Bad Observations



Dynamic Obstacles



Nonlinear dynamics

Dynamic s model

How to design the safety-assured UAV system?

[1] https://www.chinadaily.com.cn/a/202309/19/WS6508e4fca310d2dce4bb66a8.html
[2] G. Lu, W. Xu, and F. Zhang, "On-Manifold Model Predictive Control for Trajectory Tracking on Robotic Systems," *IEEE Transactions on Industrial Electronics*, pp. 1-10, 2022, doi: 10.1109/tie.2022.3212397.
[3] https://www.ehang.com/cn/
[4] https://www.alphaaitech.com/building-inspection
[5] T. Sheng *et al.*, "Unmanned Aerial Vehicle Mediated Drug Delivery for First Aid," *Adv Mater*, vol. 35, no. 10, p. e2208648, Mar 2023, doi: 10.1002/adma.202208648.
[6] The astounding athletic power of quadcopters | Raffaello D'Andrea: https://www.youtube.com/watch?v=w2itwFJCgFQ

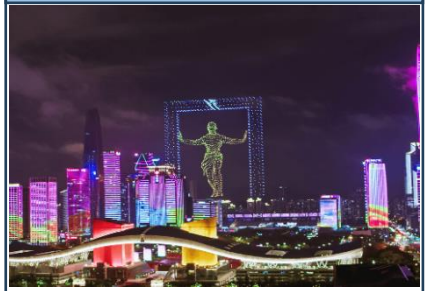
Typical applications of LAE

Transportation

Delivery

Urban Air Mobility (UAM)

Entertainment



Formation Performance

Mapping

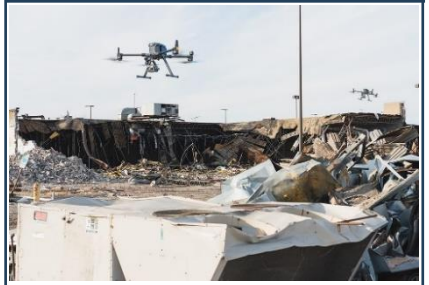
High-resolution Aerial map

Surveillance and Monitoring

Congestion Surveillance

Infrastructure Inspection

Emergency Response



Locate Victims

Telecommunication

Network Expansion

Low-altitude Economy is Coming






Upstream: Aviation Materials and Core Equipment

Key Components
- Chips
- Batteries

Aviation Materials
- Titanium Alloy
- Aluminum Alloy
- Aerospace Materials
- Special Elastomer

Core Equipment & Systems
- Mission Load
- Operating Systems
- Flight Control Systems

Midstream: Equipment Manufacturing and Supporting Services

Unmanned Aerial Vehicles (UAVs)
- Fixed-wing UAVs
- Multi-rotor UAVs
- Helicopters
- Vertical Take-off and Landing Fixed-wing UAVs

Aircraft Manufacturing
- Mission Load
- Navigation Systems
- Aviation Component Manufacturing

- Aviation Leasing
- Aviation Maintenance
- Line Maintenance
- Airframe Repair
- Aviation Power Equipment Repair
- Aviation Ground Equipment Repair
- Aviation Training
- Maintenance Training
- Pilot Training
- Airfield Construction

Downstream: Low-altitude Markets

- Low Altitude Tourism
- Low Altitude Logistics
- Low Altitude Transportation
- Low Altitude Entertainment Live Streaming
- Low Altitude Security
- Low Altitude Agricultural and Forestry Protection

Regulations

We are trying to communicate with government agencies of Hong Kong

- **13.Jun 2022**: Advanced Aviation Infrastructure Modernization Act
- **17.Oct 2022**: Advanced Air Mobility Coordination and Leadership Act
- **29.Nov 2022**: A Drone Strategy 2.0 for a Smart and Sustainable Unmanned Aircraft Eco-System in Europe
- **31.May 2023**: Interim Regulations on the Flight Management of Unmanned Aircraft
- **01.Nov 2023**: Shenzhen Special Economic Zone Low-Altitude Economic Industry Promotion Regulations
- **23.Nov 2023**: 2023 Low Altitude Economic Development White Paper (2.0)
- **Future**

Evaluation of the Market of LAE

Hong Kong has the ticket to join this game

Excessive Urban Traffic Load
Need for Efficient Transportation

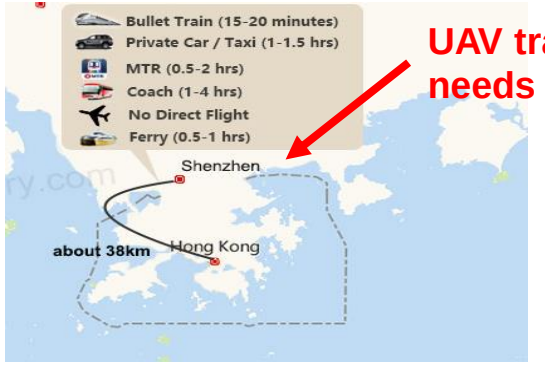
demand

Advance Productivity
Efficient Transportation

invoke

Prosperous LAE Market

Potential market	In USD	
China	$350 billion in 2022	$700 billion by 2025
US	$1 trillion by 2040	$9 trillion by 2050



UAV travel only needs 10-11 min

- Hong Kong's railway operator says average figures are between about 30,000 and 40,000 people daily when projected number is 80,000 daily
- Consumers are looking for cheaper, more convenient, more efficient transportation method

Challenges lay in the front

Regulatory Clearance

- **Noise** > Loud fan funcitoning noise
- **Safety** > Potential UAV collision
- **Privacy** > Abuse of UAV camera

Technical Issues

- **Reliability** > Reliable embeded drone system
- **Precision** > Perception Positioning Control
- **Green** > Battery sustainability
- **Failsafe** > Trust worthy failsafe system

Poor Infrastructure

- **Airport** > Take-off and landing pads
- **Recharge** > Charging station
- **Traffic** > Air traffic control mechanism

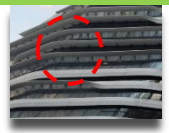
Public Acceptance

- **Noise** > UAV noise may disturb the neighborhood
- **Safety** > Collisions, drops, and so on
- **Visual polution** > Lights on UAV may bring visual polution
- **Privacy concerns** > Abuse of UAV camera may bring privacy concerns

UAV enabled building cleaning (low-altitude cleaning)

Building cleaning problem

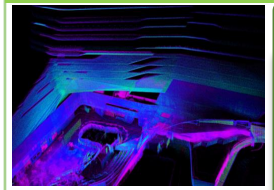
Jockey Club Innovation Tower PolyU



Protrusion

Corner

Technology reliability



Quickly reconstruct JCIT with mapping technology

Robust Control using JPCM

Multi sensor

LiDAR

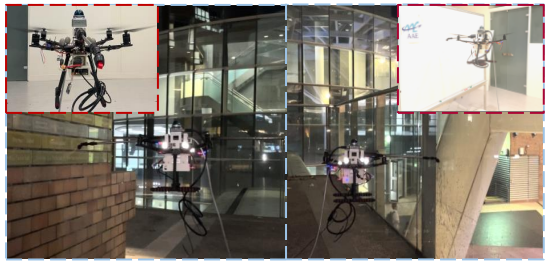
IMU

GNSS

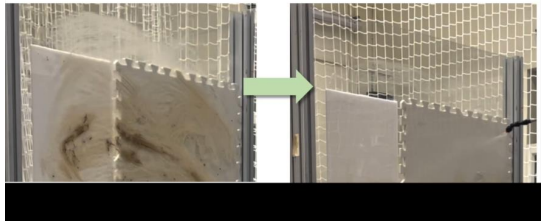
Camera

Reliability of positioning technology

Demonstrations in PolyU



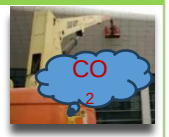
Significant cleaning effect



Traditional solution

"Spider-Man"

Cleaning robots



Machinery

- *Low efficiency (2 weeks)*
- *High energy-cost (HKD200,000)*
- *High Carbon emissions (1046.3Kg CO_2)*

UAVs-enabled building cleaning system

[1]

- Green energy (Electricity, not fuel)
- *High efficiency and flexibly*
- ***Low carbon emissions*** *(10.5 kg CO_2)*
- *Develop the LAE*

[1] https://ktvworkingdrone.com/windowcleaning/

The project is supported by **Carbon Neutrality Funding Scheme 2023/24, HKD2,000,000**

UAV enabled building cleaning (low-altitude cleaning)

UAV Design

Sensors

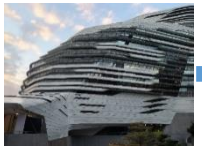
LiDAR IMU

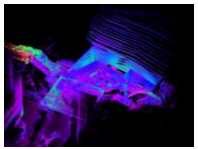
GNSS Camera

UAV cleaning solution

GLVIO (SLAM)

3D map → AI path planning



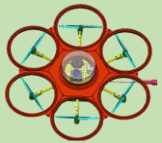


Energy and water saving

Absolute location → UAV dynamic control

UAV safe system

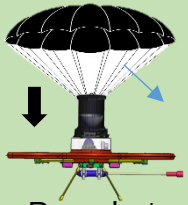
Collision Avoidance



Wing protection

Waterproof:IP65

Fall prevention



Parachute

UAV safe monitor-Multi sensor

Cleaning device

Low altitude UAV for 95% cases

Water gun

Ground

Water pump

High altitude UAV for 5% cases

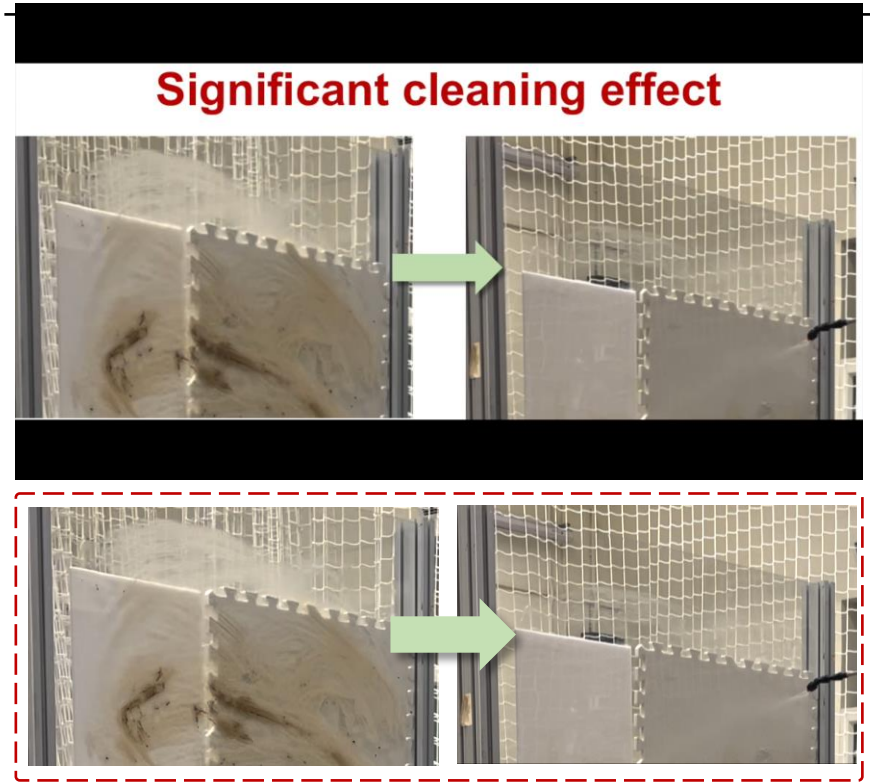
Water tank

Mount in UAV

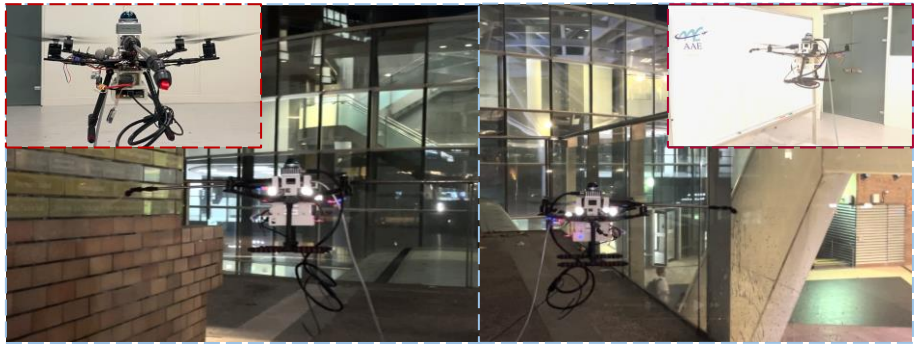
Automated replacement systems

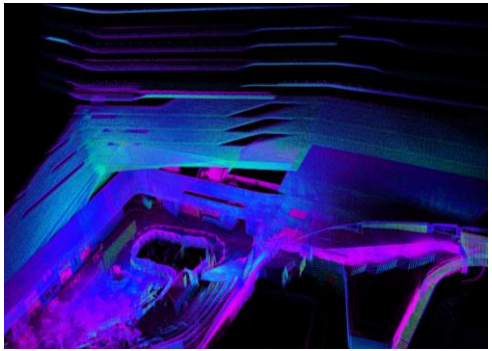
UAV building clean

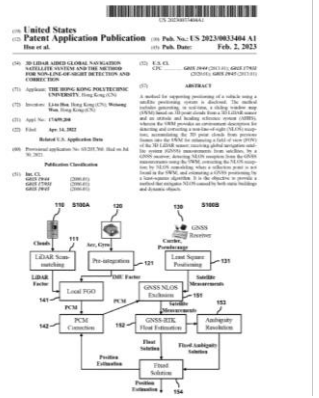
Preliminary demonstration with PolyU CFSO team



https://www.youtube.com/watch?v=m2Lm8RY2uYI





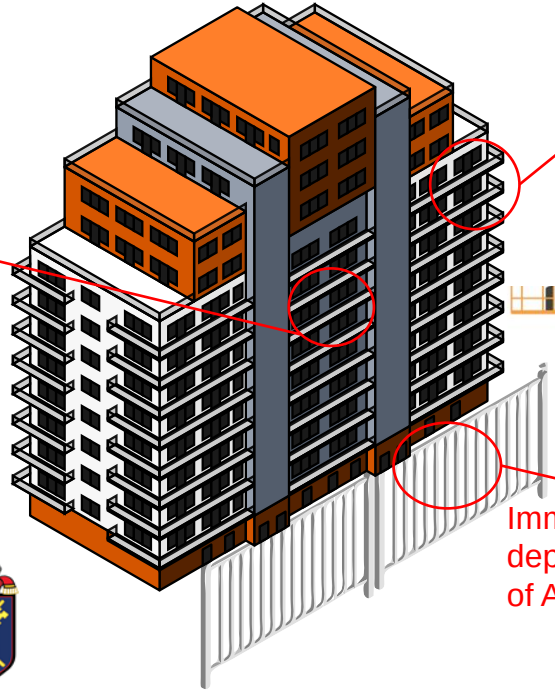


Quickly reconstruct JCIT with our technology

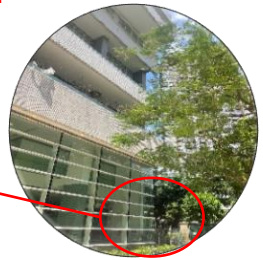
Case Study: The Providence Garden for Rehab(PGR)

Hard to clean

Unreachable



These are the common pains of most aged high-rise buildings in Hong Kong!



Immobile deployment of AWP

香 港 聖 公 會
HONG KONG SHENG KUNG HUI

SKH has invited us to test cleaning at PGR!

THE PROVIDENCE GARDEN FOR REHAB
HONG KONG SHENG KUNG HUI WELFARE COUNCIL
82 TSUN WEN ROAD, TUEN MUN, NEW TERRITORIES
TEL : (852) 3511 0700 FAX : (852) 3511 0959
E-MAIL : pgr@skhwc.org.hk
WEBSITE : http://www.skhwc.org.hk

新界屯門 眞實路八十二號 康恩園 香港聖公會福利協會

Our Ref : PGR/ADM/2024/06/05
Your Ref:

29th July 2024

UAV group of Intelligent Positioning and Navigation Laboratory
Attn: Dr. Weisong Wen

Re: Invitation Letter for Testing of " Safety-assured Unmanned Aerial Vehicles based Building Cleaning System for Energy Saving of Regular Building Maintenance" at The Providence Garden for Rehab, No.82 Tsun Wen Road, Tuen Mun, N.T., HKSAR

Dear Dr. Wen,

Hong Kong Sheng Kung Hui Welfare Council Limited (SKH), is pleased to write this letter to extend a formal invitation to Dr. Weisong Wen and the esteemed team to test the proposal, titled "***Safety-assured Unmanned Aerial Vehicles based Building Cleaning System for Energy Saving of Regular Building Maintenance***", at The Providence Garden for Rehab (PGR) in mid-August. We will prepare the test field and are looking forward to the prototype test.

As you know, we are critically concerned with the safety of all participants and pedestrians, please inform us ahead with what else protocols needs to be prepared and followed. We are expecting the feedback and hope the test result may help us to achieve more in charitable deeds.

Should you have any queries on this matter, please feel free to contact the undersigned at 35110800. For the site operation issue, please contact Mr. Gilbert Ko at 35110844.

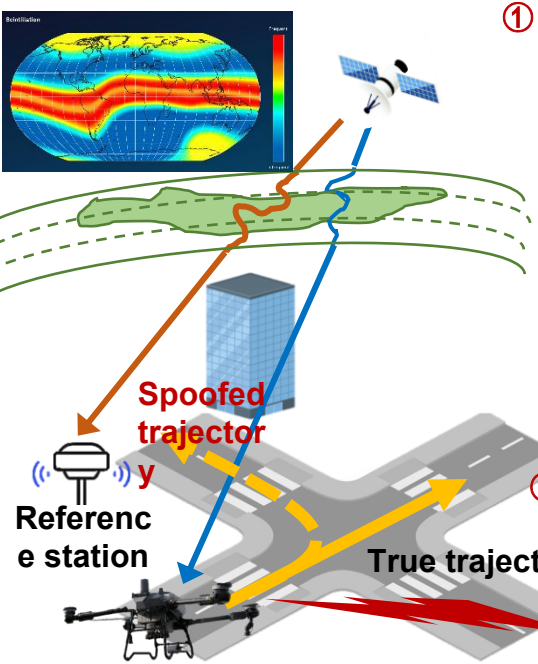
Yours Sincerely,

Vincent HU
Service Director

Operated by
Hong Kong Sheng Kung Hui Welfare Council Limited
(由香港聖公會福利協會有限公司營運)
A MEMBER AGENCY OF THE COMMUNITY CHEST

Precise and Robust Positioning is Essential for UAS Development

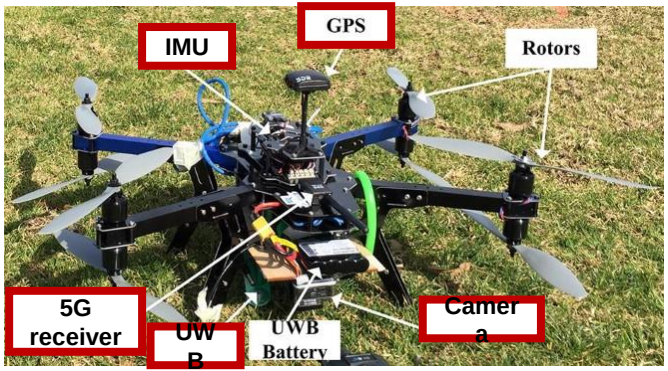
Challenges of GNSS positioning in urban areas



① **Ionospheric error**

- Ionospheric model error
- Ionospheric scintillation
- Ionospheric storm

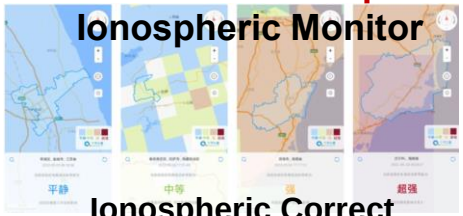
Ionosphere

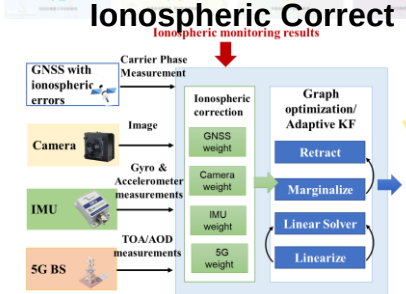


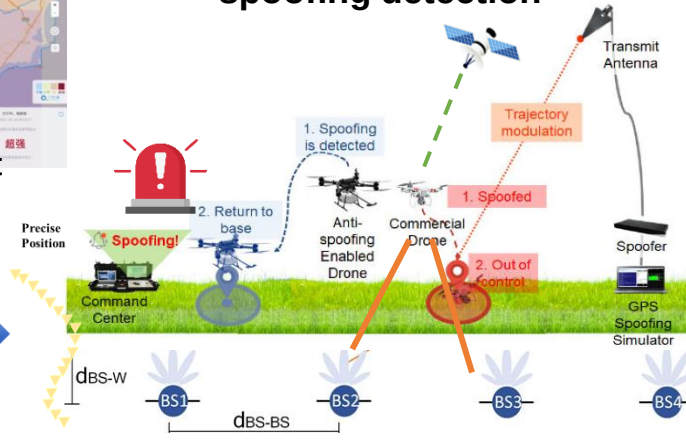
② **Spoofing attack**

The accuracy and robustness of GNSS are significantly degraded in complex urban conditions

Multi-sensor-based Ionospheric error & spoofing detection solution

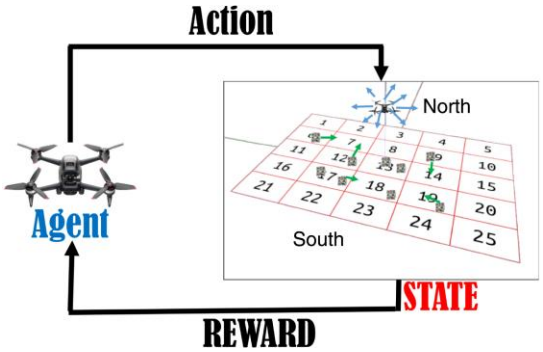






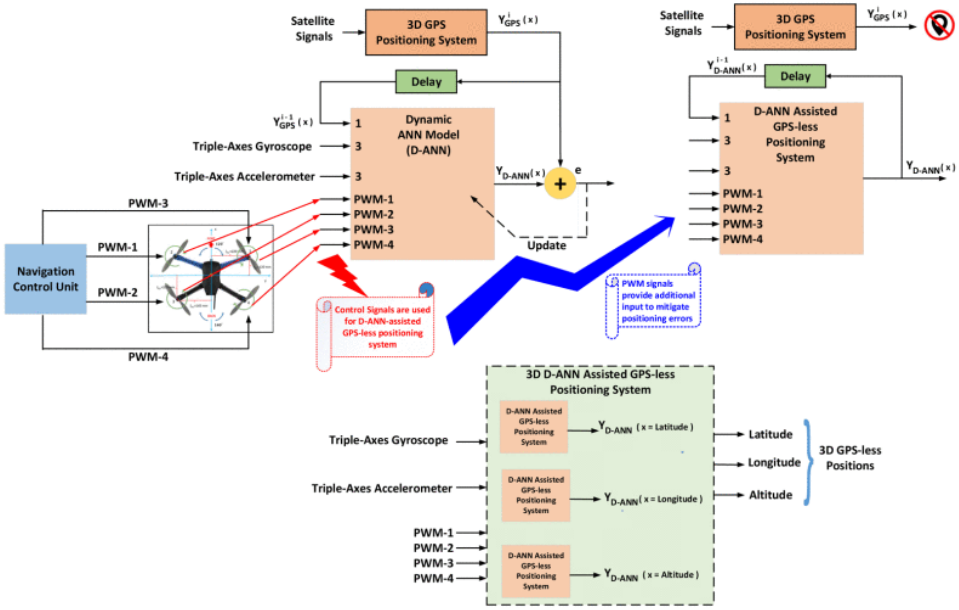
What can AI help with drones?

- **Adaptive Path Planning**



- **Object detection of drones**

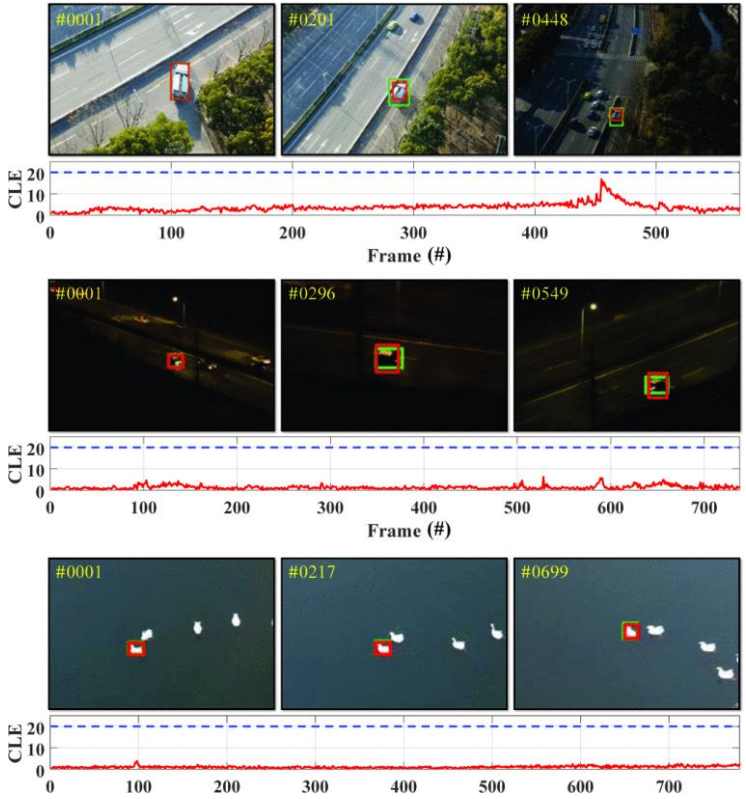
- **AI-based drones for GPS navigation**

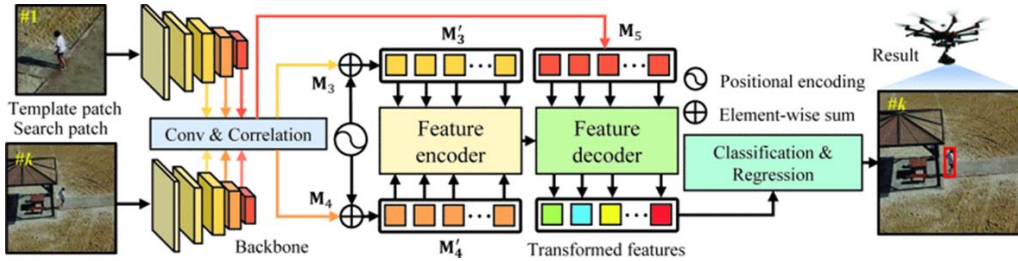


AI-based UAV navigation framework with digital twin technology for mobile target visitation

Dynamic Artificial Neural Network-Assisted GPS-Less Navigation for IoT-Enabled Drones

Transformer for Efficient Aerial Tracking



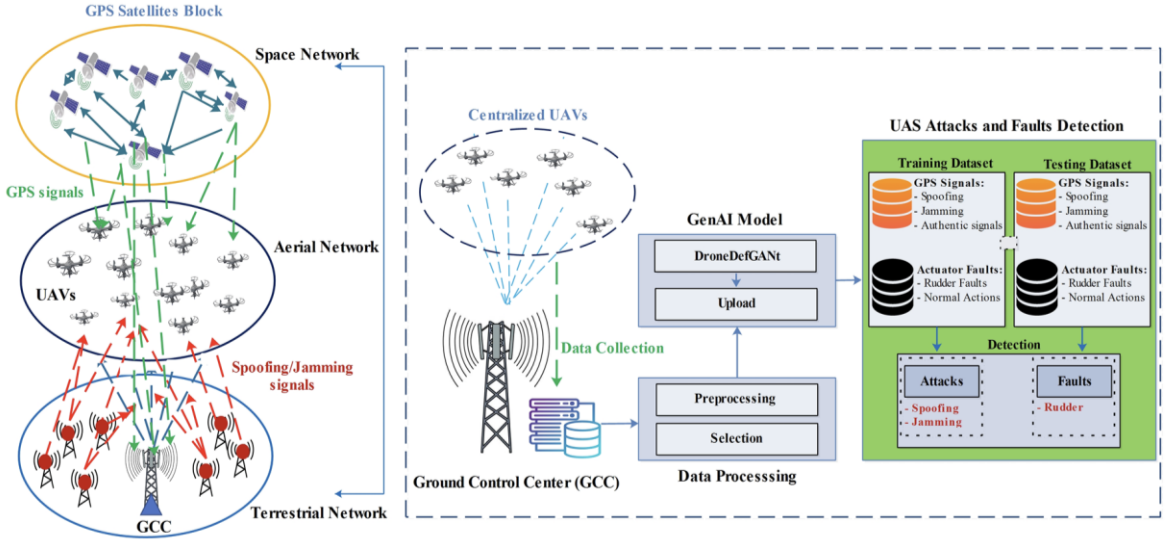


The tracking results and ground truth are marked with red and green boxes. The CLE score below the blue dotted line is considered as the success tracking result in the real-world tests.

https://ieeexplore.ieee.org/document/9710895

DroneDefGANt for Detecting UAS Attacks and Faults

GenAI can *detect and prevent cyberattacks* by continuously learning and adapting to new threats and vulnerabilities.



DroneDefGANt, a GenAI-based GAN and transformer models.

The DroneDefGANt is to detect both external UAS attacks like GPS spoofing and jamming, and internal attacks such as actuator faults.

DroneDefGANt: A Generative AI-Based Approach for Detecting UAS Attacks and Faults.

Tools for learning AI

Development environment (computer)

- *Python is an interpreted, object-oriented, high-level programming language with dynamic data types.*

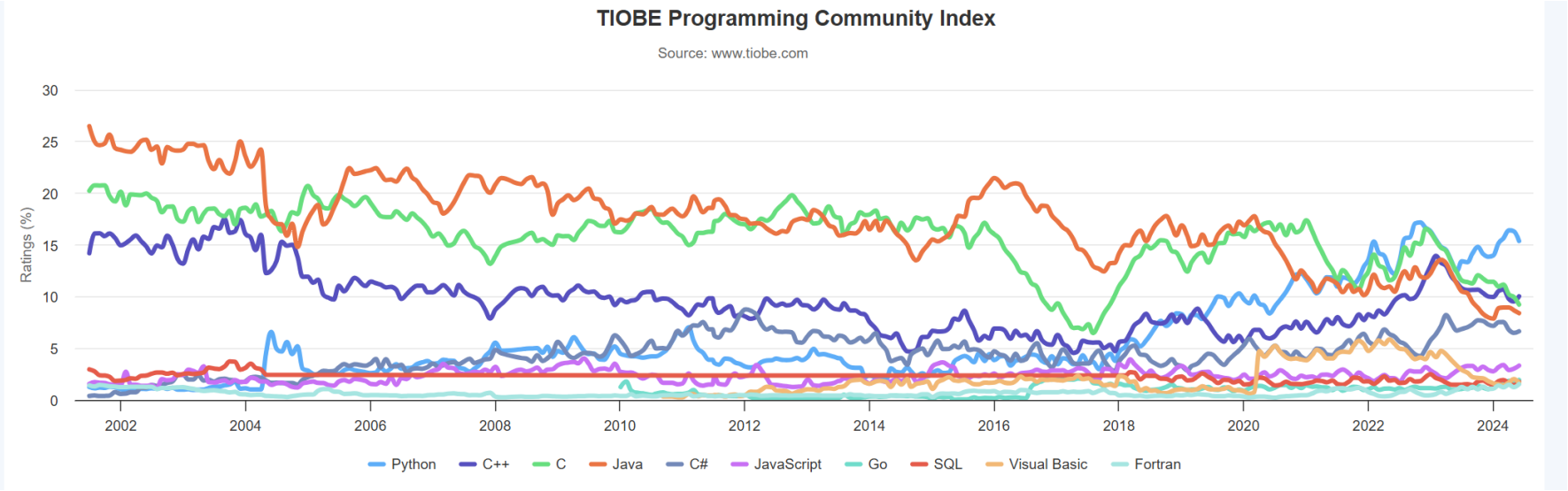
- *Open-source package and environment management system that runs on Windows, macOS, and Linux. Install, run, and update packages and their dependencies.*

- *Free software, open standards, and web services for interactive computing across all programming languages.*

Python Index Top!

TIOBE https://www.tiobe.com/tiobe-index/

Jun 2024	Jun 2023	Change	Programming Language
1	1		Python
2	3	▲	C++
3	2	▼	C
4	4		Java
5	5		C#



Anaconda Distribution

Millions of developers and data scientists research, innovate, and discover using Anaconda Distribution.

Repository
Over 8,000 open-source data science and machine learning packages, Anaconda-built and compiled for all major operating systems and architectures.

Learn More >

Using the Jupyter Notebook

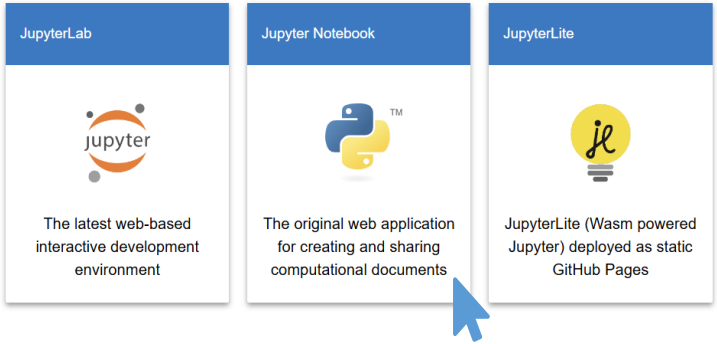
Jupyter Notebook: The Classic Notebook Interface

The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience.

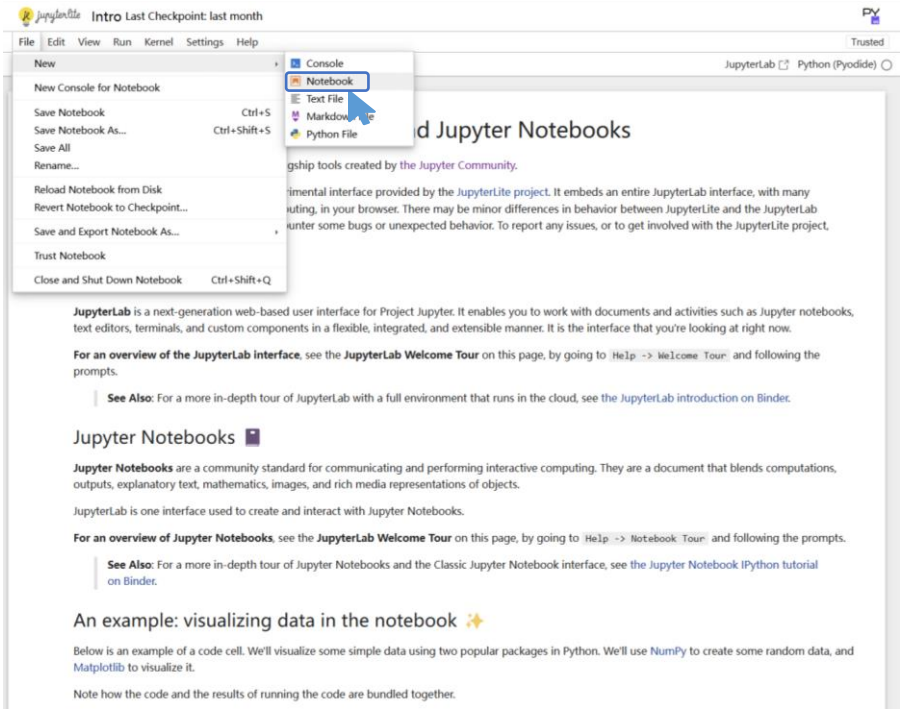
Free software, open standards, and web services for interactive computing across all programming languages

Jupyter notebook

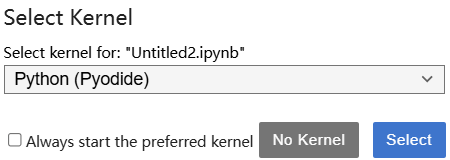
1. https://jupyter.org



2. New file Notebook (.ipynb)



3. Select python kernel

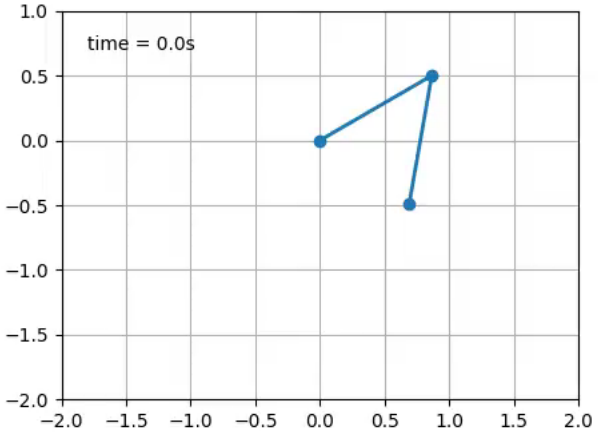


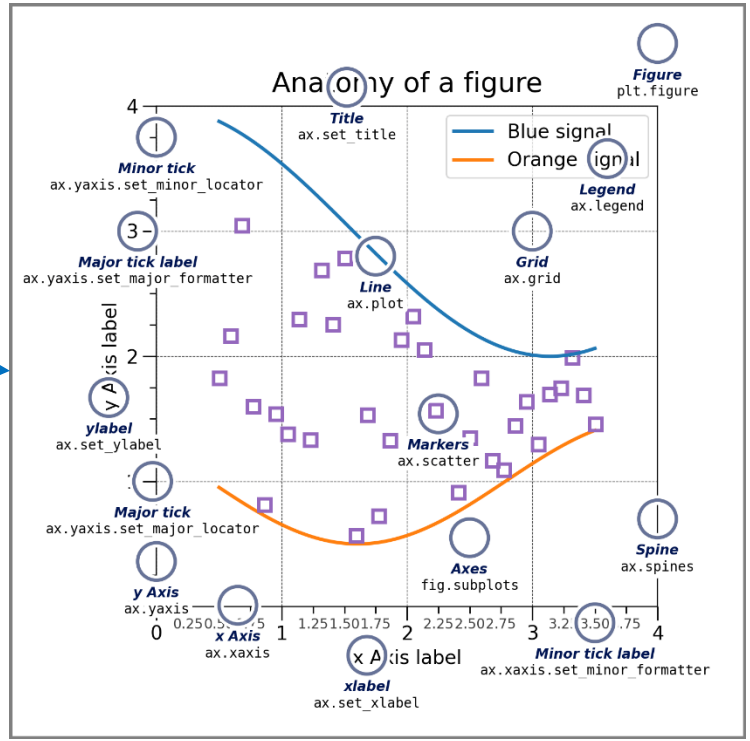
Basic Toolkit for learning AI

matplotlib

*Python **basic plotting library**, a few lines of code to generate plots, histograms, bar charts, scatter plots and more.*

Example: the double pendulum problem.

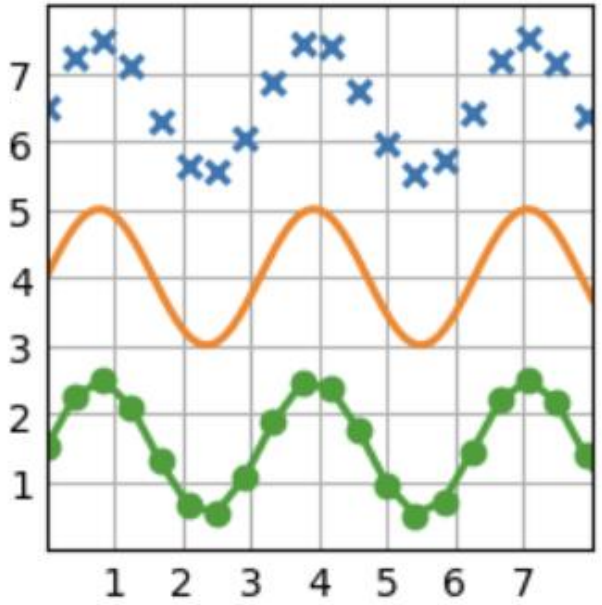




[Source]

matplotlib

```
import matplotlib.pyplot as plt
import numpy as np

plt.style.use('_mpl-gallery')

# make data
x = np.linspace(0, 10, 100)
y = 4 + 1 * np.sin(2 * x)
x2 = np.linspace(0, 10, 25)
y2 = 4 + 1 * np.sin(2 * x2)

# plot
fig, ax = plt.subplots()

ax.plot(x2, y2 + 2.5, 'x', markeredgewidth=2)
ax.plot(x, y, linewidth=2.0)
ax.plot(x2, y2 - 2.5, 'o-', linewidth=2)

ax.set(xlim=(0, 8), xticks=np.arange(1, 8),
       ylim=(0, 8), yticks=np.arange(1, 8))


plt.show()
```



Pandas

*A powerful toolset for **analyzing structured data** that can be used to quickly implement data in/out, indexing.*

Pandas provides two types of classes for handling data:

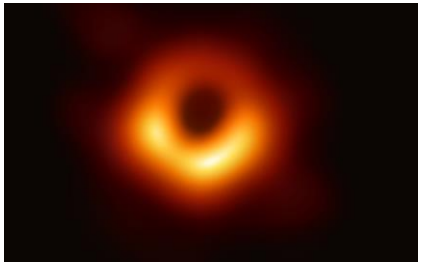
- Series: a *one-dimensional labeled array* holding data of any type such as *integers, strings, Python objects etc.*
- DataFrame: a *two-dimensional data structure* that holds data like a two-dimension *array or a table with rows and columns.*

Numpy

*Basic package for scientific computing with python. At its core, it is based on array arithmetic with **N-dimensional data objects ndarray**.*

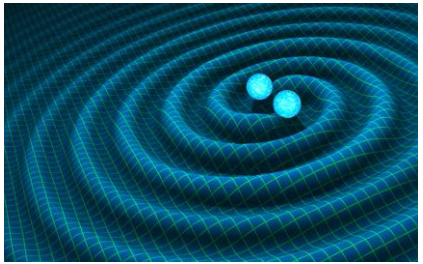
CASE STUDIES

First Image of a Black Hole



How NumPy, together with libraries like SciPy and Matplotlib that depend on NumPy, enabled the Event Horizon Telescope to produce the first ever image of a black hole.

Detection of Gravitational Waves



In 1916, Albert Einstein predicted gravitational waves; 100 years later their existence was confirmed by LIGO scientists using NumPy.

Sports Analytics

Cricket Analytics is changing the game by improving player and team performance through statistical modelling and predictive analytics.

Pose Estimation using deep learning

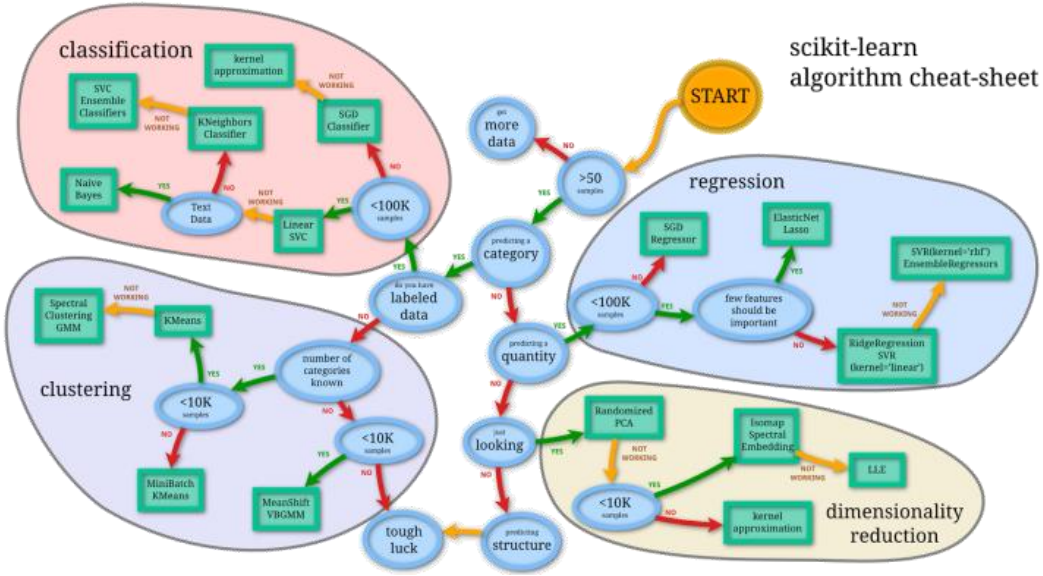
DeepLabCut uses NumPy for accelerating scientific studies that involve observing animal behavior for better understanding of motor control, across species and timescales.

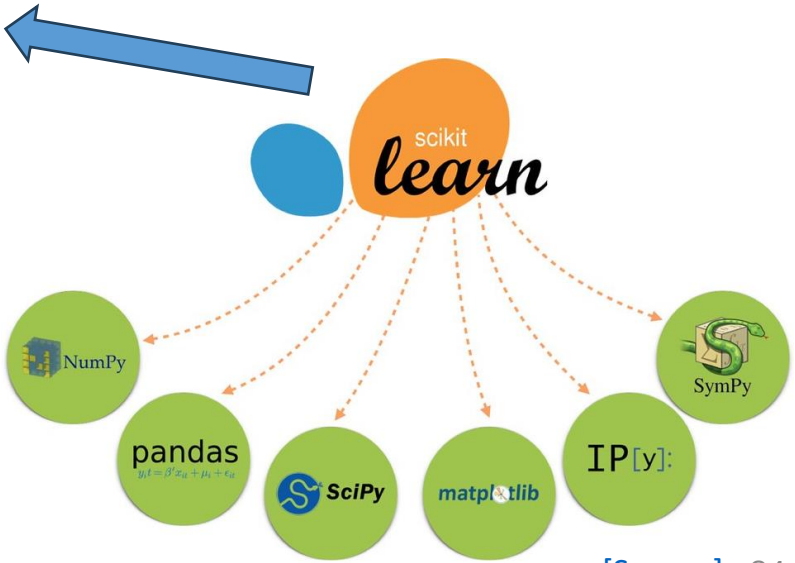
[Source]

Scikit-learn

Python's most powerful machine learning library

which is based on NumPy, SciPy and matplotlib, and supports a variety of machine learning models, including classification, regression, clustering and dimensionality reduction.





[Source]

Keras

Keras is a deep learning API written in Python and capable of running on top of either JAX, TensorFlow, or PyTorch.

First contact with Keras

The core data structures of Keras are **layers and models**. The simplest type of model is the Sequential model, a linear stack of layers.

Here is the model:Sequential

```
import keras
model = keras.Sequential()
```

Once your model looks good, configure its learning process with :.compile()

```
model.compile(loss='categorical_crossentropy',
              optimizer='sgd',
              metrics=['accuracy'])
```

Stacking layers is as easy as :.add()

```
from keras import layers

model.add(layers.Dense(units=64, activation='relu'))
model.add(layers.Dense(units=10,
activation='softmax'))
```

Common AI Framework

Pytorch

PyTorch was developed by Meta AI (formerly Facebook AI Research Lab) and is built on the Torch library. It was first released in 2016 and quickly gained attention for its flexibility, ease of use, and dynamic computational graphs.

Tensorflow

Launched in 2015, TensorFlow grew out of Google's internal research into its products and services. It evolved from an earlier framework called DistBelief and is designed to be more flexible and efficient.

Whether you favor the flexibility and user-friendliness of PyTorch or the scalability and robustness of TensorFlow, your choice will be a critical step in your AI and ML endeavors.

Thank you for your attention.

Dr Weisong Wen
If you have any questions or inquiries, please feel free to contact me.
Personal Page: https://polyu-taslab.github.io/team/
Email: welson.wen@polyu.edu.hk