EE4016
AI with Deep Learning

Master Deep Learning to Lead in the AI Era

Prof. PO Lai-Man

Department of Electrical Engineering
City University of Hong Kong
http://www.ee.cityu.edu.hk/~lmpo/

EE4016 Course Website

- **URL Addresses:**
  - Homepage: https://www.ee.cityu.edu.hk/~lmpo/ee4016/index.html
  - Schedule Webpages: https://www.ee.cityu.edu.hk/~lmpo/ee4016/schedule.html
- **Visit the course website regularly for downloading course materials**
  - Course website provides schedule, lecture notes, assignments, and project details
  - Submission of assignments and projects is done via the CANCAS system
- Use the following credentials to access course materials:
  - Username: **students**
  - Password: **ai2026**

EE4016 Course Description

- AI with Deep Learning is a hands-on course that equips aspiring AI scientists and engineers with the skills to design, train, and deploy deep neural networks using PyTorch.
- Through collaborative Python projects and real-world datasets, you'll master key architectures—MLPs, CNNs, RNNs, Transformers and LLMs—alongside optimization, regularization, and hyperparameter tuning.
- Blending theory with practice, the course empowers you to **build scalable AI solutions and innovate at the forefront of deep learning research and application**.

Outline

- **Today**
  - **Course Overview and Administration**
    - Teaching Staff, Textbook, Website, Topics, Grading, Assignments, **Project**
  - **History of AI with Deep Learning**
  - **Math Review: Linear Algebra, Calculus, and Basic Probability**
- **Next Lecture**
  - Perceptron and Multilayer Perceptron (MLP)

Teaching Staff

- Course Instructor
  - **Prof. PO Lai-Man（布礼文）**
  - eelmpo@cityu.edu.hk
  - Room G6506, Green Zone, 6/F, AC1
  - Phone: 3443-7779

- Teaching Assistants (TAs)
  1. Mr. LIU Yuyang
     - yuyangliu5-c@my.cityu.edu.hk
  2. Mr. WU Haoxuan
     - haoxuanwu2-c@my.cityu.edu.hk

Communication

- **Canvas Q&A:**
  - Students are encouraged to post questions and actively participate in discussions on Canvas QA portal.
  - Use Canvas for all technical questions and public communication with the course staff.
- **For private questions email to me or TAs**
  - eelmpo@cityu.edu.hk
- **Course announcements will be post on Canvas**
  - Make sure you check it regularly

Grading of EE4016

- 15% : **3 Assignments with programming exercises** (Week 5, 9, 13)
  - Questions and Programming assignments in Colab
  - Assignments take lots of time. Start early!!
- 20% : **Group Project**
  - Proposal (Week 4)
  - Oral Presentation (Week 12 or 13)
  - Final Report, PPT, source code, demo video (Week 14)
- 15% : **Quiz** 5% (Week 6) and **Midterm Exam** 10% (Week 11)
- 50% : **Final Exam** (2 hours)

Remarks:

- *To pass the course, students are required to achieve at least 30% in course work and 30% in the final exam*
- ***Understanding the different between** collaboration and academic infraction*

Main Topics of the Course

- **Course Overview**
- **History of AI with Deep Learning**
- **Review of Linear Algebra, Calculus and Probability**
- **Multi-Layer Perceptron (MLP)**
  - Perceptron and MLP
  - Activation function, Loss function
  - Backpropagation and Gradient Descent
  - Regularizations and Optimizations
- **Convolutional Neural Networks (CNNs)**
  - Convolutional Neural Layer
  - LeNet, AlexNet, VGGNet, InceptionNet, ResNet, DenseNet, EfficientNet
  - Computer Vision Applications:
    - Object Detection, Image Segmentation
- **Recurrent Neural Network (RNN)**
  - Introduction to NLP
  - Tokenization and Word Embeddings
  - Vanilla RNN, LSTM, GRU
  - Encoder-Decoder Models and Attention
- **Transformers and Large Language Models (LLMs)**
  - Self-Attention Mechanism
  - Transformers
  - BERT, GPT, GPT-2, GPT-3, T5 and BART
  - LLM Decoding and Prompt Engineering
  - Model Quantization
  - Parameter-Efficient Fine-Tuning (PEFT)
- **Preference Alignment and Multimodality**
  - Preference Alignment: Instruction-Tuning, RLHF, DPO
  - Multimodality: CLIP, GPT-4, LLaVA

EE4016 Prerequisites

- **Prerequisites:**
  - Linear Algebra, Multi-variable Calculus and Probabilities, **and**
  - Object-Oriented Programme (e.g. Python)

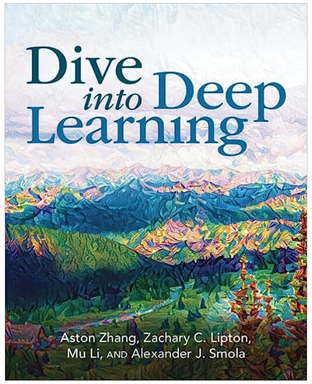
WARNING

- Please be advised, **EE4016 is a course with a STRONG mathematical and programming components.**
  - **Focus**: Deep learning architectures trainable via gradient-based approaches.
    - Linear Algebra and Matrix Calculus.
    - Probability and Statistics.
  - **Skills gained through assignments and projects**:
    - Proficiency in Python and familiarity with deep learning libraries like PyTorch.

Readings

- Reading materials will be assigned to each class. They will be listed on the schedule webpage of the course website and include medium articles and some classic research papers.
  - https://www.ee.cityu.edu.hk/~lmpo/ee5438/schedule.html
- It is recommended to read the designated sections before class. Be prepared to discuss the topic intelligently and/or ask questions about material you do not understand.

Reference Books


Dive into Deep Learning
Aston Zhang, Zachary C. Lipton,
Mu Li, AND Alexander J. Smola

- **Dive into Deep Learning**
  - Aston Zhang, Zack Lipton, Mu Li, and Alex Smola. https://d2l.ai/
- **Deep Learning** by Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
  - https://www.deeplearningbook.org
- **Patten Recognition and Machine Learning (PRML)** by Christoper C. Bishop.
  - https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf
- **Natural Language Processing with Distributed Representations** by Kyunghyun Cho.
  - https://arxiv.org/abs/1511.07916

Canvas Submissions

- All assignments and projects must be submitted in electronic formats such as MS-Words, PDF and source code.
- Soft copies such as pdf files and related documents need to be **submitted to Canvas by 11:00pm** on the due date.
- **Submission format:**
  - To make it easier for graders to process your electronic submissions, please submit them using the following filename format:
  - Assignment_Number_Student_Name_Student_Number.pdf
    - Assignment01_Chan_Chi_Ming_501234565.pdf
    - Project_Proposal_Group01.pdf
      - For group project submission, each team only needs to submit one copy to CANVAS.

Late Submission Policy

- All assignments and reports mush be uploaded to CANVAS before 11:00PM on the due date.
  - NO late assignment is accepted without previous arrangement with the instructor.
  - If approved, **late submission receives 20% per business day penalty**.
- Students may work together on the assignment, but copying is unacceptable.
- **Work to learn. Don’t work for marks**

Cheating and Attendance

- In particular copying your fellow classmate's assignments or programs, is a very serious offense!
  - If you are found cheating, you will automatically get an F grade in this course, and your act will be reported to the Department for necessary disciplinary actions.
  - Please don't let others copy your assignments or programs as we don't have a way to tell who is copying who and you may be liable to the penalties.
- **Class attendance is required**, and you are held responsible for any material or announcements.

Introduction to AI, ML and DL

DL ⊂ NN ⊂ ML ⊂ AI

Artificial Intelligence (AI)

- The broad concept of machines performing tasks that typically require **human intelligence**.

Machine Learning (ML)

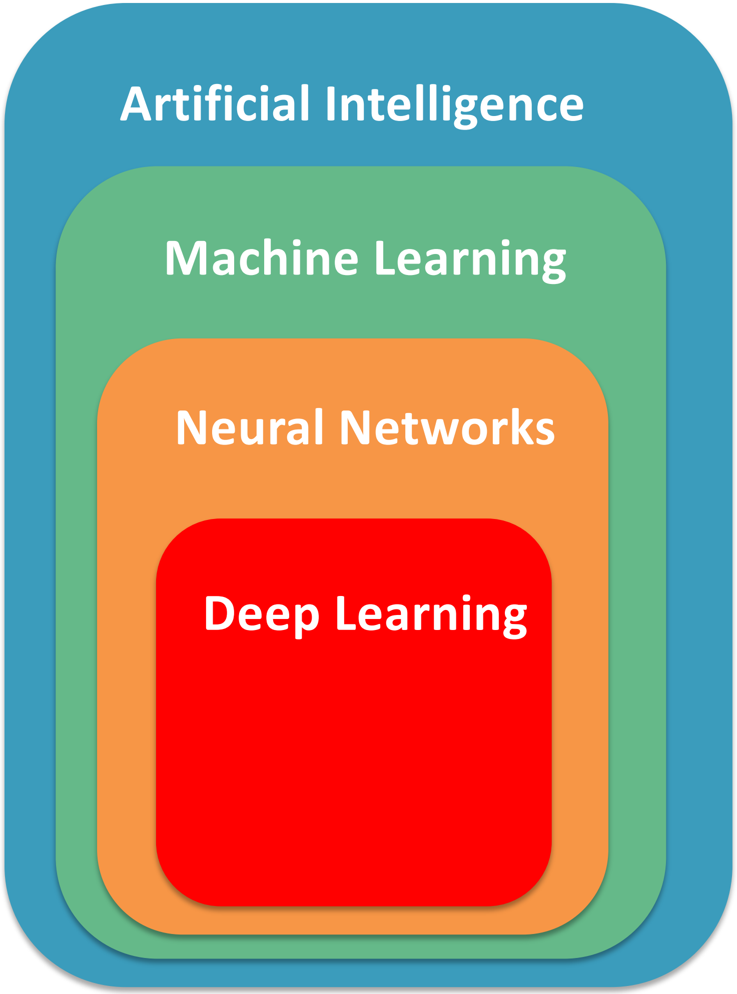
- A subset of AI where systems **learn from data**, identify patterns, and make decisions with minimal human intervention.

Neural Networks (NN)

- A subset of ML. **Computational models inspired by the human brain**; used in ML to recognize complex patterns.

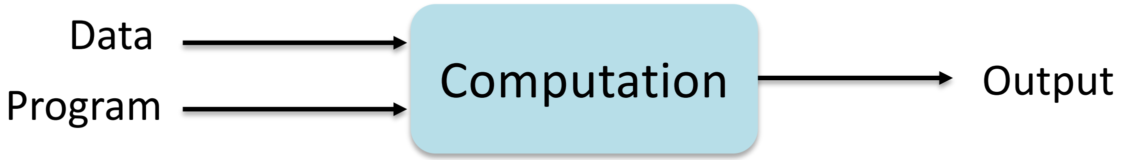
Deep Learning (DL)

- A subset of NN using **neural networks with many layers** ("deep" architectures).

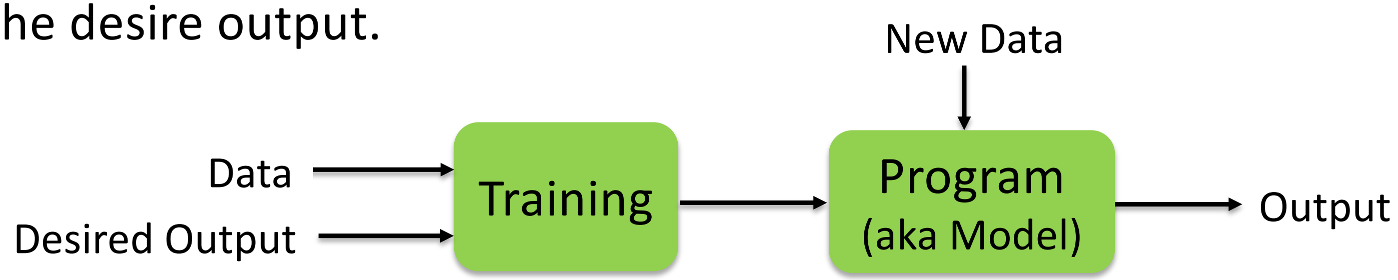


Machine Learning

Traditional Programming: Developers write rules (program) that produce an output.



Machine Learning: Developers write a **training algorithm**, that finds rules, which produce the desire output.



Machine learning **is a subfield of AI** that trains algorithms to learn from data in order to make predictions without being explicitly programmed.

Machine Learning Approaches

- **Supervised Learning**
  - Uses **labeled data**: Each data point is paired with an output label that guides the learning process.
  - The model learns to map inputs to outputs based on these labeled examples.
- **Unsupervised Learning**
  - Uses **unlabeled data**: The model must find patterns and structure on its own, without any specific guidance.
  - The goal is to discover hidden patterns, groupings, or dimensionality reductions within the data.
- **Reinforcement Learning**
  - **No explicit instructions**: The model learns through trial and error, receiving feedback in the form of rewards or penalties.
  - The model's goal is to maximize cumulative rewards by learning a sequence of successful actions, balancing exploration and exploitation.

Semi-Supervised and Self-Supervised Learnings

- **Semi-Supervised Learning**
  - Combines a **small amount of labeled data** with a **much larger set of unlabeled data**: The model initially learns from the limited labeled data and then leverages the unlabeled data to improve its performance and generalization.
- **Self-Supervised Learning**
  - Employs **unlabeled data** but generates its own **pseudo-labels** by creating pretext tasks or objectives derived from the data itself: The model learns to predict these pseudo-labels, enabling it to capture meaningful representations of the data.

Tow Main Types of Machine Learning

Supervised Learning

- **Data**: $(\mathbf{x}, y)$
  - $\mathbf{x}$ is data, y is label
- **Goal**: Learn function to map

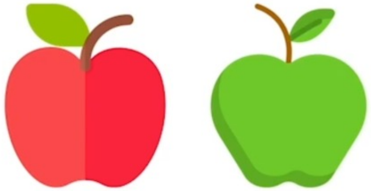
$$\mathbf{x} \rightarrow y$$

- **Example**:

This thing is an apple

Unsupervised Learning

- **Data**: $\mathbf{x}$
  - $\mathbf{x}$ is data, no labels
- **Goal**: Learn underlying structure (distribution)
- **Example**:

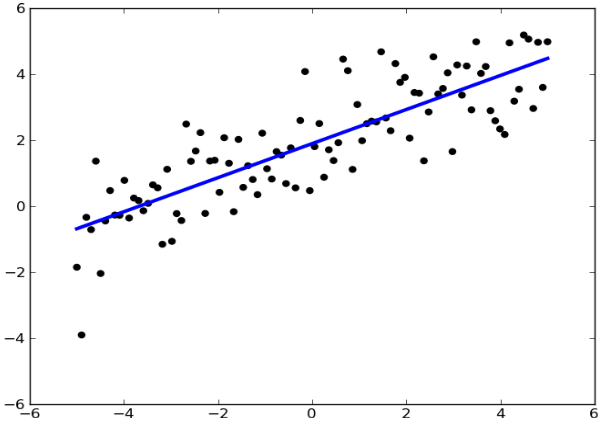


These two things are similar

Two Main Machine Learning Problems

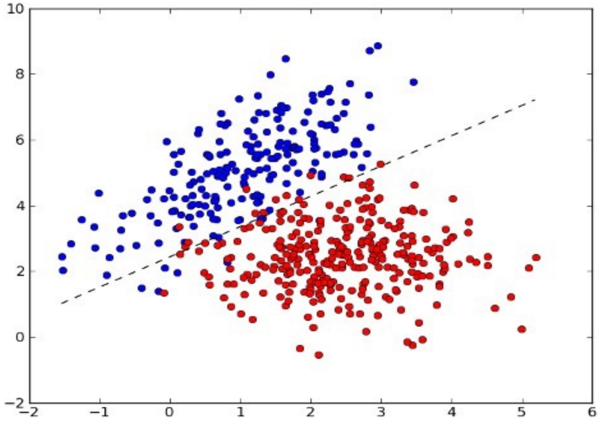
- Broadly speaking we can break most ML problems down into two categories:

Regression



In regression problem, we try to infer a function that **maps continuous inputs to continuous output**.

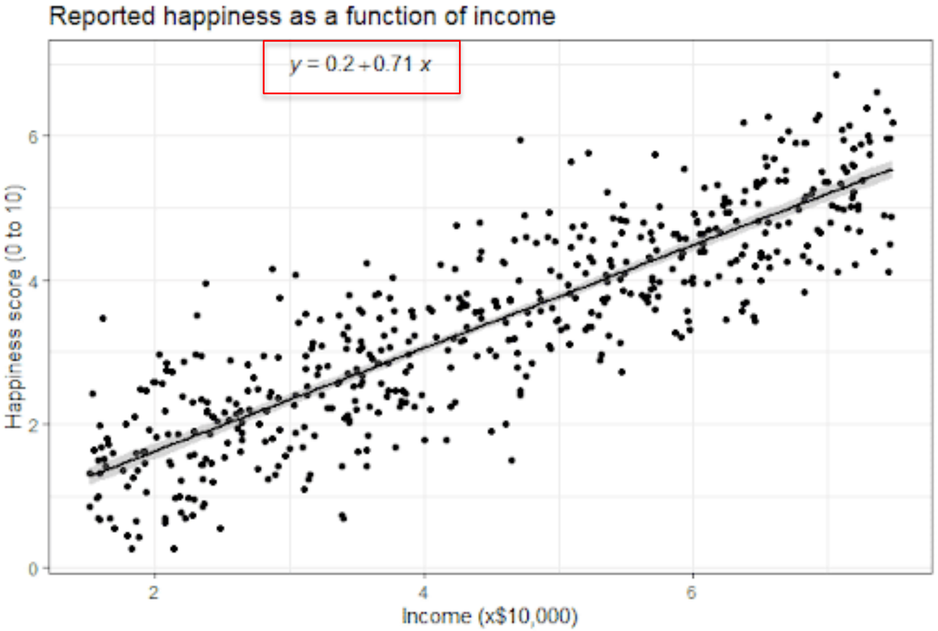
Classification



In classification problem, we try **to assign a discrete label to each input point.**

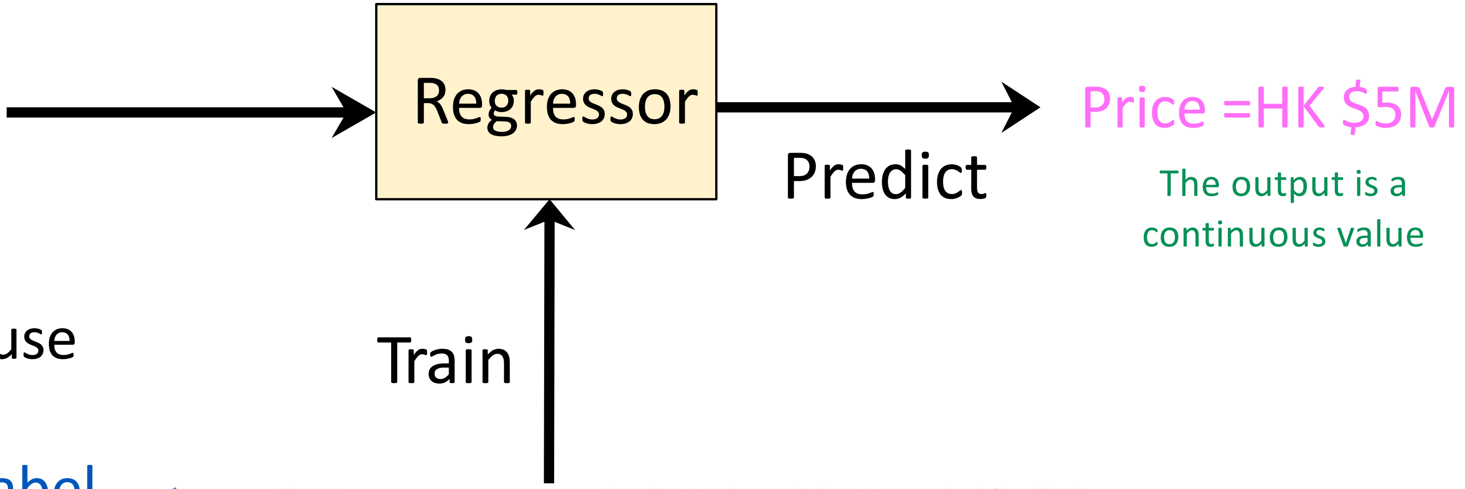
Regression Example: Happiness vs Income

- Using linear regression, **it is found that a significant relationship (p < 0.001) between income and happiness (R2 = 0.71 ± 0.018)**, with a 0.71-unit increase in reported happiness for every 10,000 increase in income.



Regression: Housing Price

Features of a House

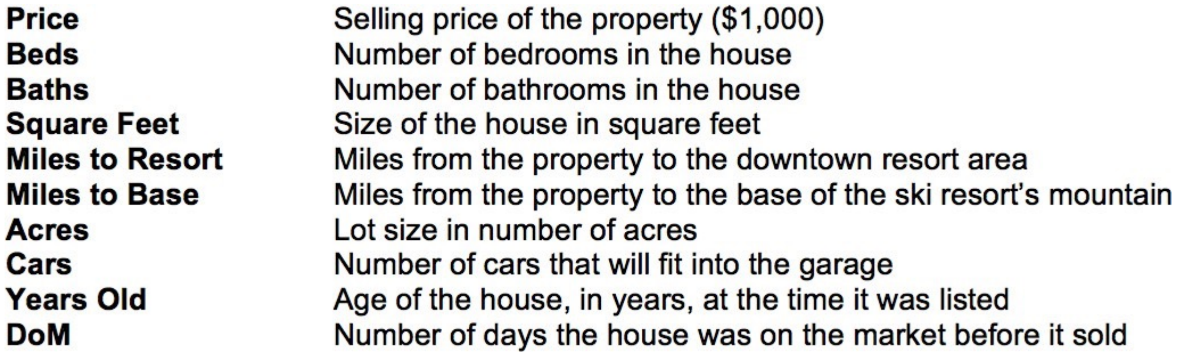


Price =HK $5M

The output is a continuous value

label

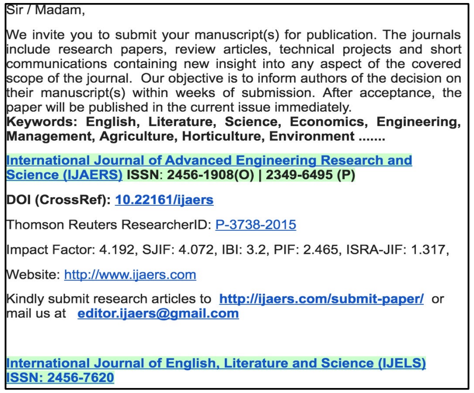
features



Price	Selling price of the property ($1,000)
Beds	Number of bedrooms in the house
Baths	Number of bathrooms in the house
Square Feet	Size of the house in square feet
Miles to Resort	Miles from the property to the downtown resort area
Miles to Base	Miles from the property to the base of the ski resort's mountain
Acres	Lot size in number of acres
Cars	Number of cars that will fit into the garage
Years Old	Age of the house, in years, at the time it was listed
DoM	Number of days the house was on the market before it sold

Classification Example: Spam Detection

Binary classification



Sir / Madam,

We invite you to submit your manuscript(s) for publication. The journals include research papers, review articles, technical projects and short communications containing new insight into any aspect of the covered scope of the journal. Our objective is to inform authors of the decision on their manuscript(s) within weeks of submission. After acceptance, the paper will be published in the current issue immediately.
Keywords: English, Literature, Science, Economics, Engineering, Management, Agriculture, Horticulture, Environment ....

International Journal of Advanced Engineering Research and Science (IJAERS) ISSN: 2456-1908(O) | 2349-6495 (P)

DOI (CrossRef): 10.22161/ijaers

Thomson Reuters ResearcherID: P-3738-2015

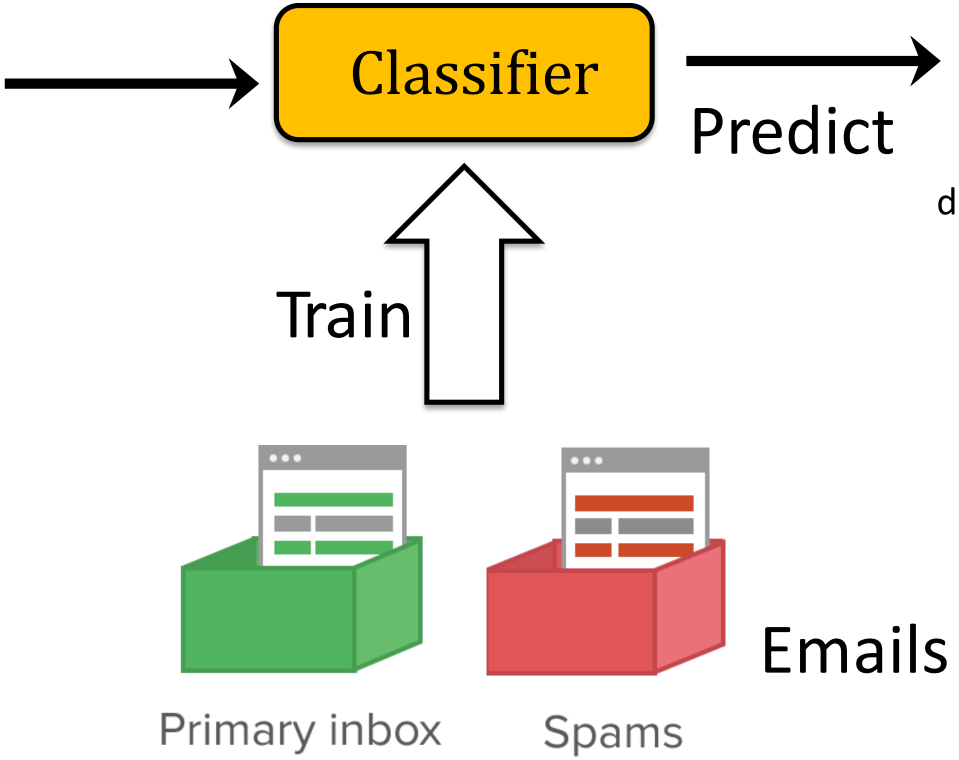
Impact Factor: 4.192, SJIF: 4.072, IBI: 3.2, PIF: 2.465, ISRA-JIF: 1.317,

Website: http://www.ijaers.com

Kindly submit research articles to **http://ijaers.com/submit-paper/** or mail us at **editor.ijaers@gmail.com**

International Journal of English, Literature and Science (IJELS) ISSN: 2456-7620

New Email



Classifier

Predict

Train

Is Spam

The output is a discrete value: 0 or 1

Emails with Labels

Classification: Face Recognition

Classifier

Predict

The output is a discrete value

“Matt Damon”

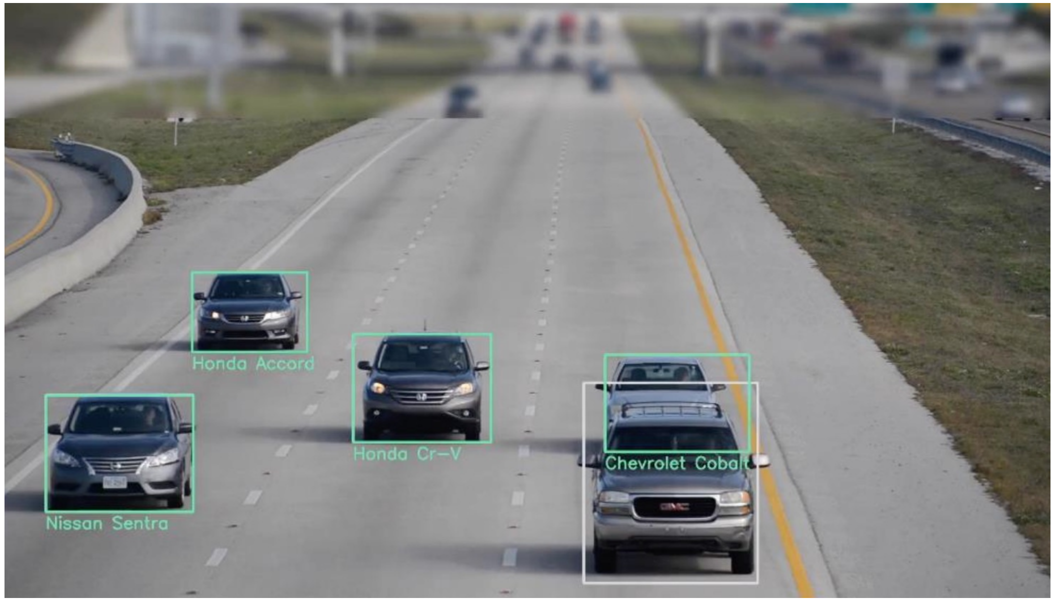
Train

Multi-class Classification

Faces & Names

Object Recognition (Regression + Classification)

- Object recognition **requires to solve both regression and classification**
  - **Regression**: identify the object locations in x and y coordinates (continue variables)
  - **Classification**: identify the object inside the bounding box (discrete labels)



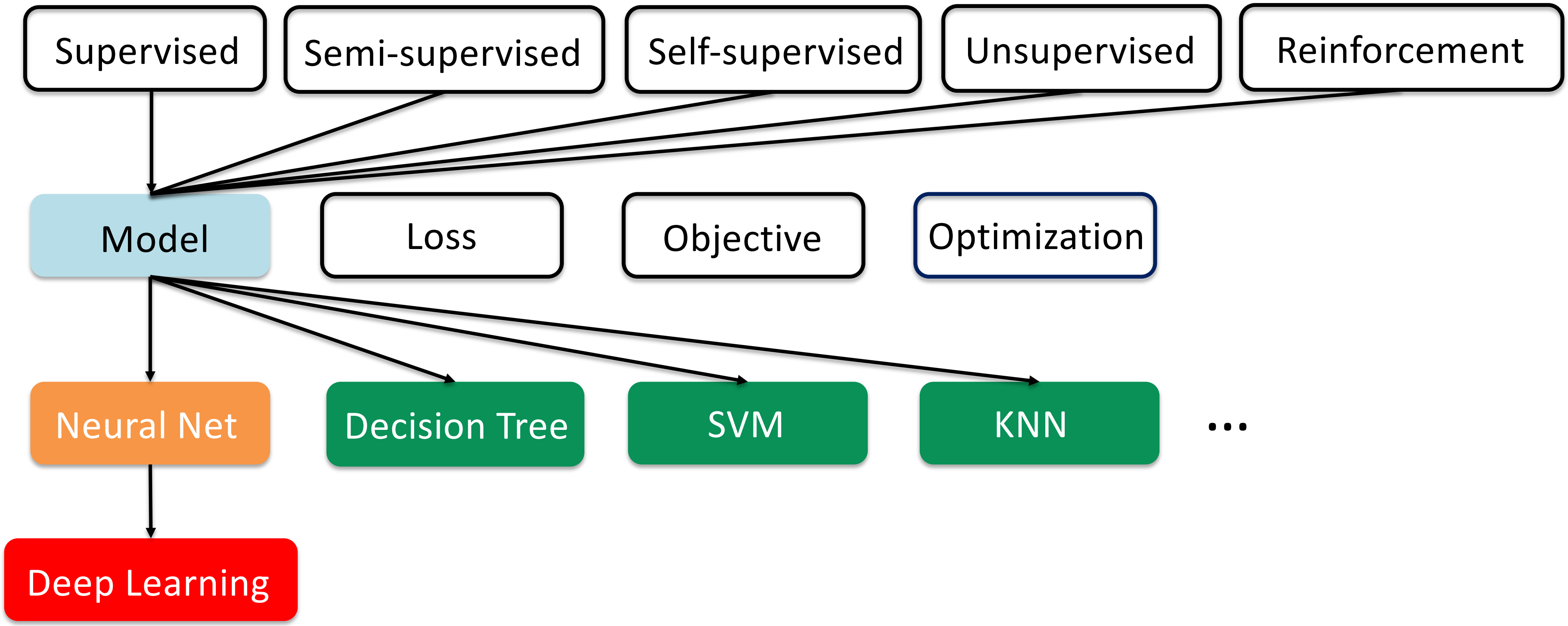
Components in Machine Learning Training

- **Model**: Output predicts from inputs
  - e.g. listing house => **model** => sale price
- **Loss**: Measure **difference** between predicts and ground truth labels
  - e.g. Square Error = (predict_price – sale_price)^2
- **Objective**: Any function to optimize during training
  - e.g. minimize the sum of losses over all examples
- **Optimization**: Learn **model parameters** by solving the objective function

Types of Machine Learning Models

- **Linear Regression**: Predicts the value of a dependent variable based on one or more independent variables by fitting a straight line to the data.
- **Logistic Regression**: Use a sigmoidal functions to perform binary classification
- **Decision Trees**: Use trees to make decisions for solving classification problem
- **K-Nearest Neighbors (KNN)**: predicts the value of a new data point based on the similarity to its nearest neighbors in the training set.
- **Support Vector Machine**: It finds an optimal decision boundary to classify data or predict values by maximizing the margin.
- **Neural Networks**: Use neural networks to learn feature representations for regression and classification problems

Neural Network ⊂ Machine Learning

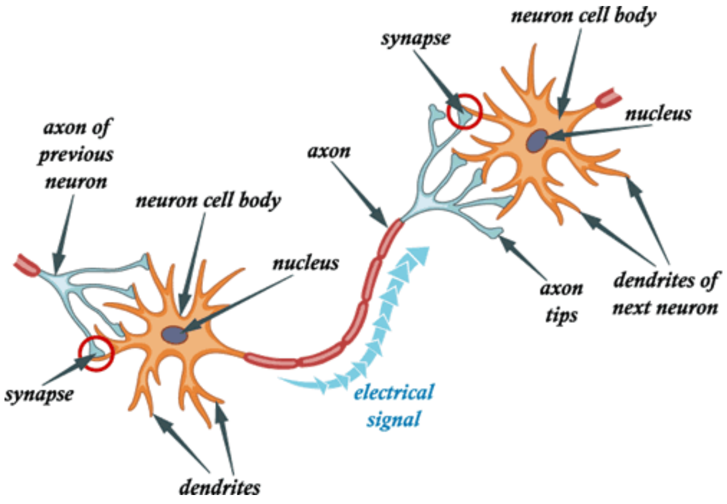


Deep Learning

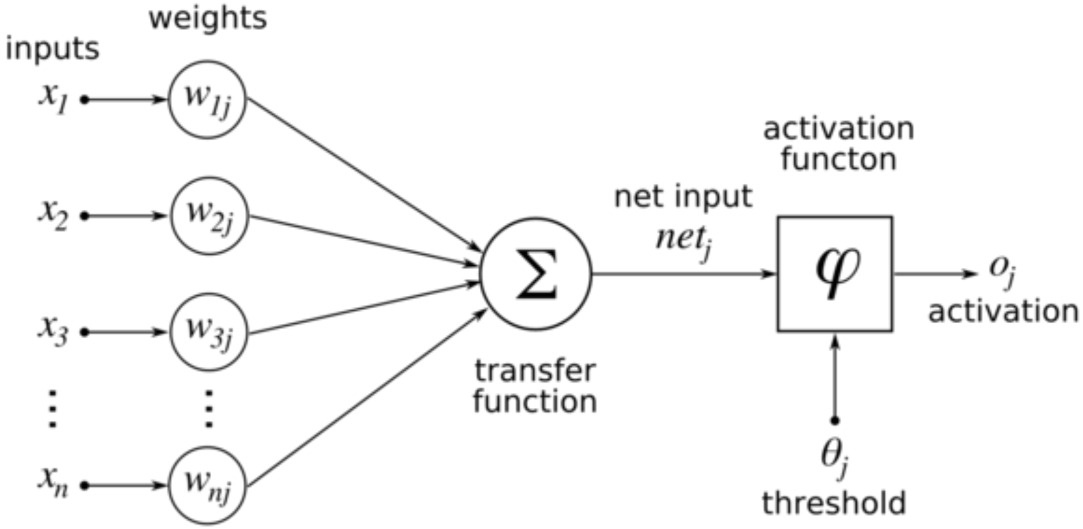
- **Deep learning** can be defined as the use of multi-layered **neural networks** with differentiable programming to perform machine learning
  - Deep Learning is a subfield of Neural Networks
  - **Deep learning** uses neural networks with many layers **(at least 3 layers)** to achieve outstanding performance on complex machine learning tasks like face recognition, object detection, and language translation.
  - **Differentiable Programming** because neural networks are differentiable functions that can be optimized using gradient-based methods like backpropagation.

The Building Blocks of Neural Networks?

- Neural networks consist of processing elements (**neurons**) and connections between them.



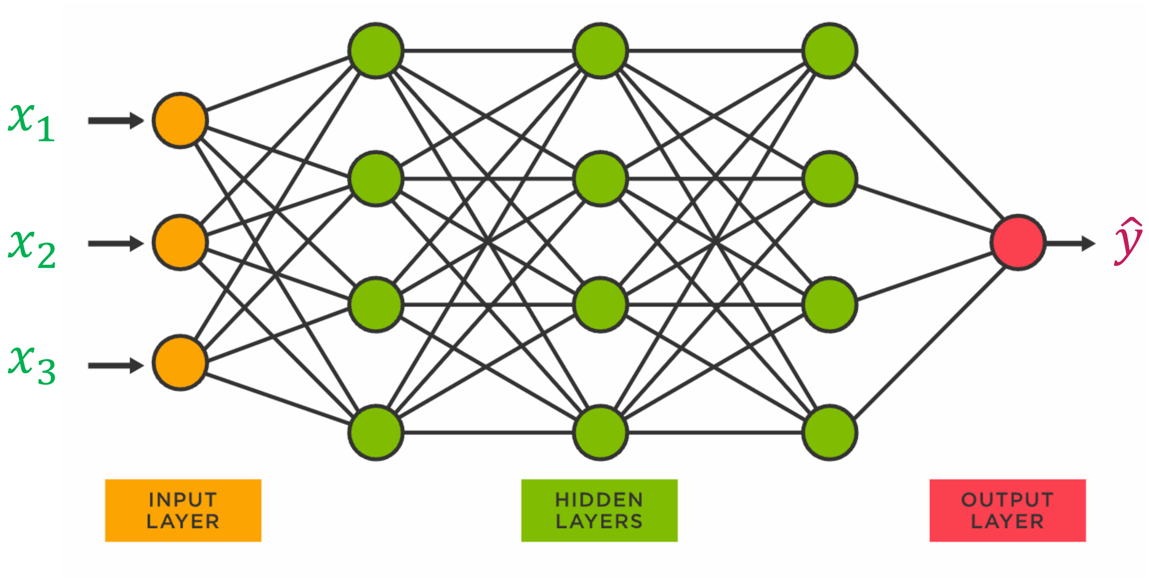
Structure of Typical Neuron



Structure of Artificial Neuron

Neural Networks

- Neural networks (NNs) are computing systems inspired by the biological neural networks that constitute human brains.

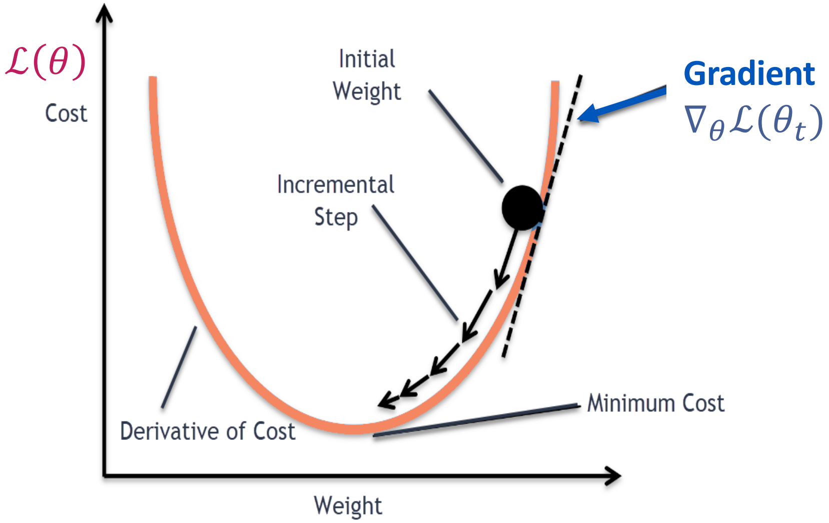


$$\hat{y} = f_\theta(x_1, x_2, x_3)$$

https://www.tibco.com/reference-center/what-is-a-neural-network

What is Differential Programming?

- To identify the weights of the neural network, we need to training it with data and the key idea behind the neural network training is **Gradient Descent**.
- We know from calculus that a function’s gradient always points in the direction of steepest increase and that we can use that directional information to minimize the function



Python Program

Cost Function

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell\left(y^{(i)}, f_\theta\left(x^{(i)}\right)\right)$$

Compute Gradient: $\nabla_\theta \mathcal{L}(\theta)$

Update Parameters θ:

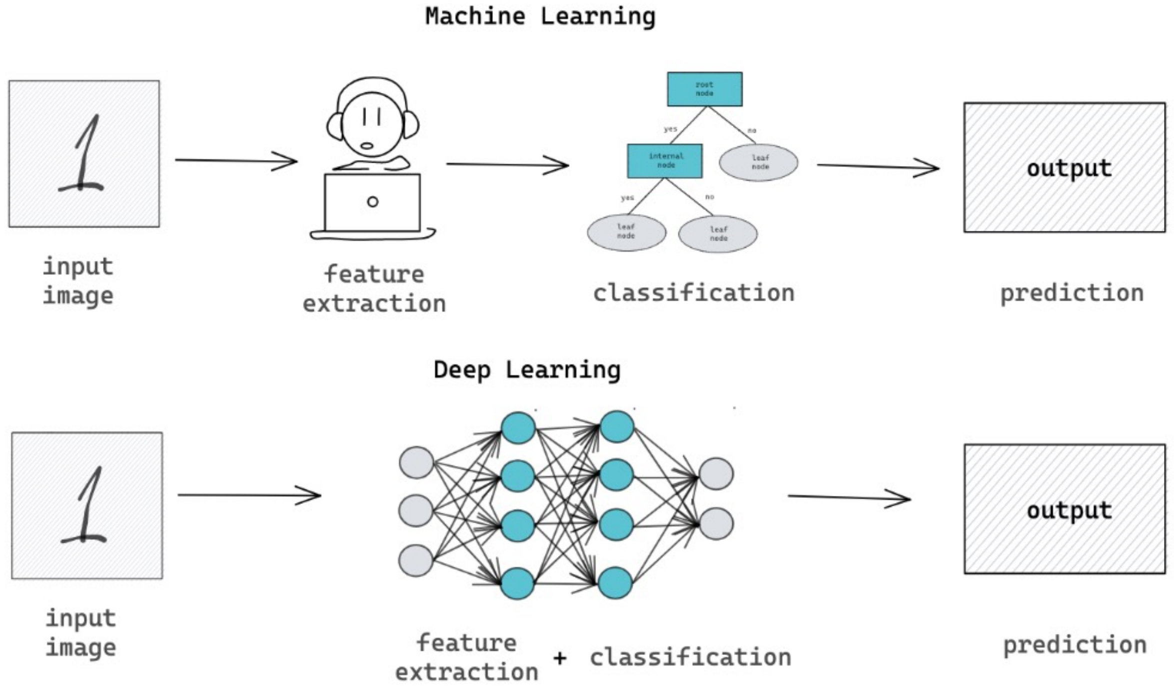
$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_\theta \mathcal{L}(\theta_t)$$

Why Deep Learning Now?

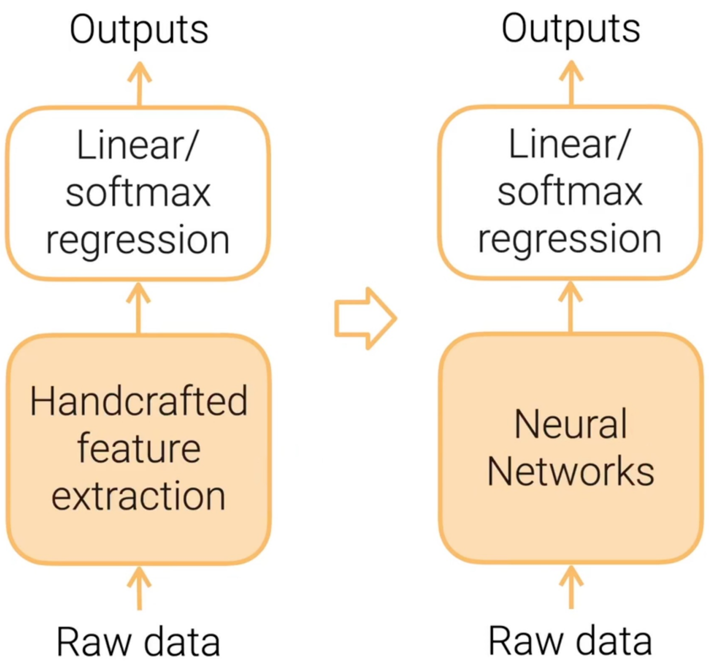
- In the last two decades, deep learning (DL) has replaced many traditional AI and ML techniques in various applications.
  - **Multi-layer perceptron (MLP)** models have replaced linear/logistic regression, decision trees, and SVMs in general machine learning.
  - **CNNs and Transformers** are now standard for computer vision tasks like as image classification, object detection, image segmentation, etc.
  - **RNNs and Transformers** have replaced HMMs in natural language processing, achieving state-of-the-art performance in machine translation (Google Translate) and text generation with Large Language Models (LLMs) like ChatGPT.
- **Deep Learning** has revolutionized machine learning by enabling **highly accurate and efficient models** for a wide range of applications.

End-to-End Architectures of Deep Learning

- **No handcrafted feature engineering** in deep learning



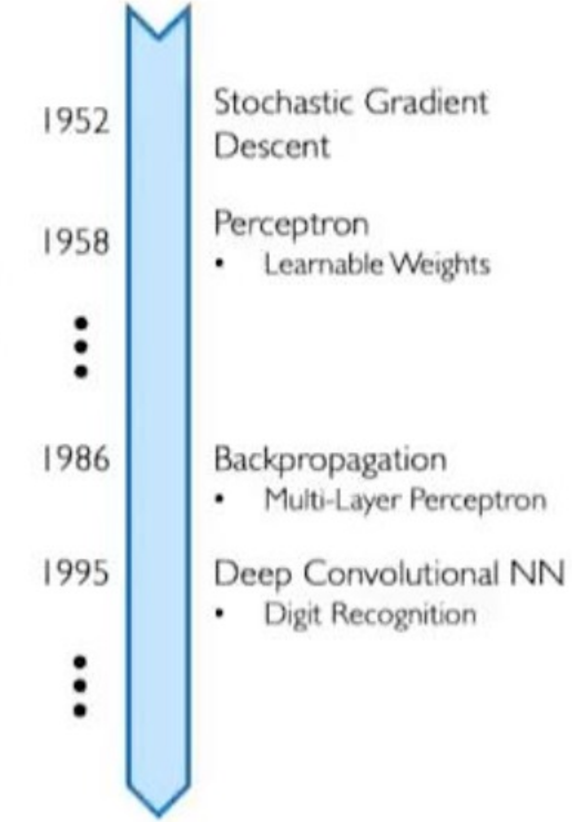
Handcrafted Features => Learned Features


Outputs
Linear/ softmax regression
Handcrafted feature extraction
Raw data
Outputs
Linear/ softmax regression
Neural Networks
Raw data

- Deep neural networks Learning usually requires more data and more computation
- Deep neural network architectures to model data structures
  - **Multilayer Perceptron (MLP)**
  - **Convolutional Neural Network (CNN)**
  - **Recurrent Neural Network (RNN)**
  - **Transformer**
  - **Deep Reinforcement Learning (DRL)**

What Factors Enable Deep Learning now?

- Neural Networks data back decades, so why the resurgence?



1952 Stochastic Gradient Descent

1958 Perceptron

- Learnable Weights

⋮

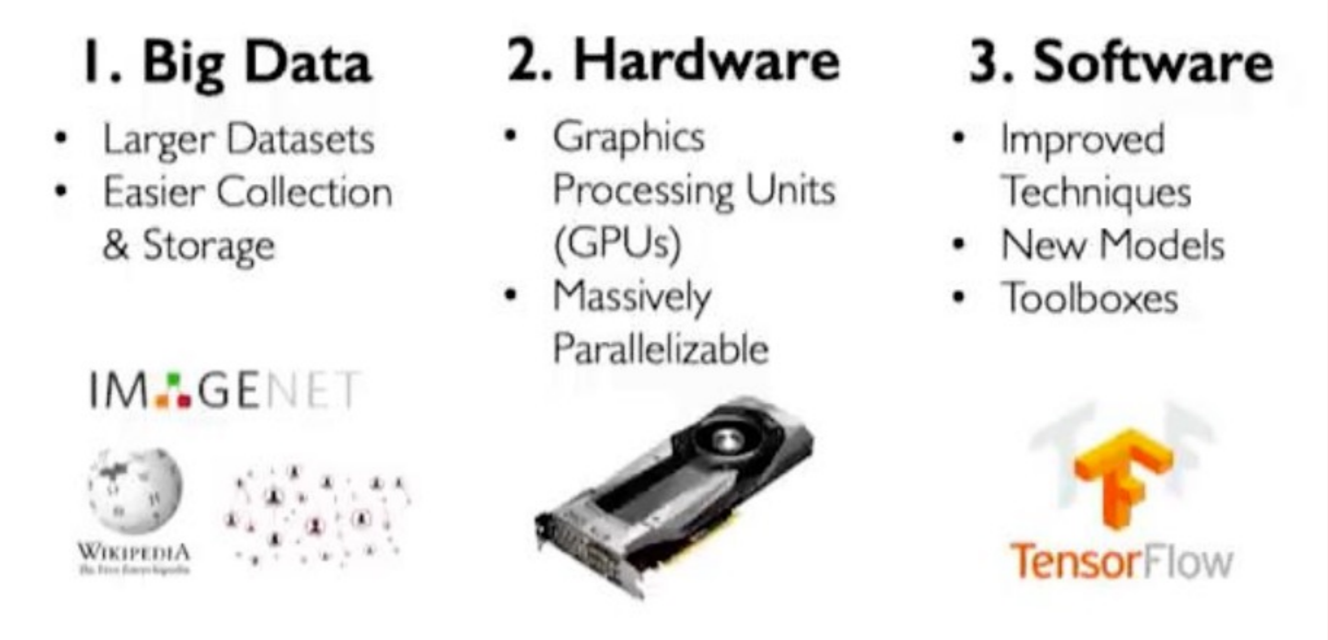
1986 Backpropagation

- Multi-Layer Perceptron

1995 Deep Convolutional NN

- Digit Recognition

⋮



1. Big Data

- Larger Datasets
- Easier Collection & Storage

IMAGENET
WIKIPEDIA
The Free Encyclopedia

2. Hardware

- Graphics Processing Units (GPUs)
- Massively Parallelizable

3. Software

- Improved Techniques
- New Models
- Toolboxes

TensorFlow

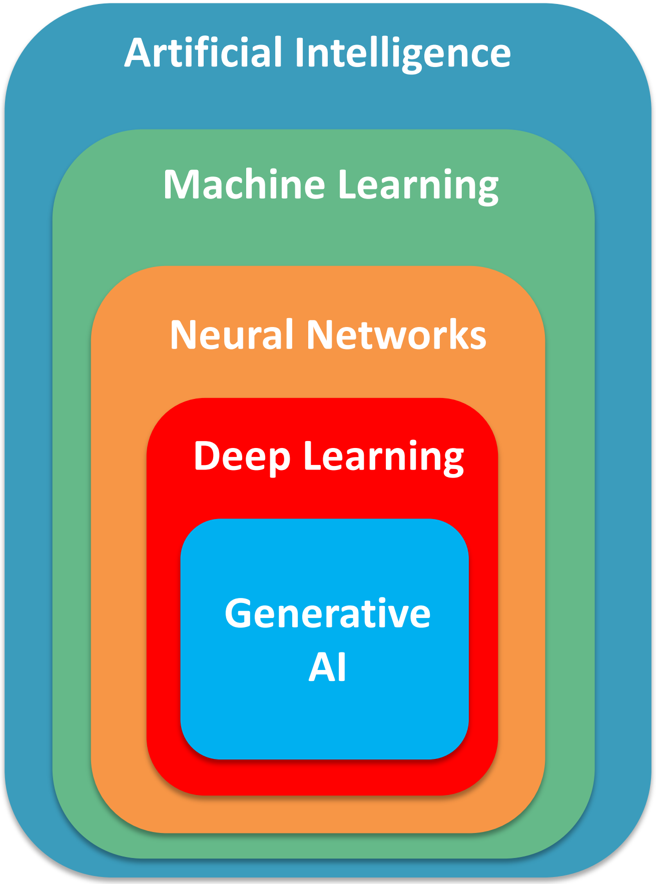
Breakthroughs in AI with Deep Learning

- **1943**: McCulloch & Pitts Neuron
- **1957**: Perceptron (Single layer Neural Network)
- **1960**: Delta Learning Rule
- **1969**: XOR problem (Limitations of the perceptron)
- **1970s**: Multilayer Perceptron (MLP)
- **1986**: Backpropagation
- **1989**: UAT (1 hidden layer is good enough)
- **2006**: RBM initialization (breakthrough)
- **2009**: GPU
- **2010**: breakthrough in Speech Recognition (Dahl et al., 2010)
- **2012**: breakthrough in ImageNet (Krizhevsky et al. 2012)
- **2015**: "superhuman" results in Image and Speech Recognition
- **2016**: AlphaGo "superhuman" results in Go playing
- **2022**: ChatGPT "human-level" results in diverse domains
- **2023**: GPT-4 Turbo and Gemini with Multimodality

AI, ML, NN, DL, Generative AI

Generative AI

- Generative AI creates new content, such as text, images, or videos, by learning from large datasets. It has many applications.
- Examples:
  - **Large Language Models** (LLMs): ChatGPT, GPT-5, Gemini 2.5 Pro, Claude 4 Sonnet, DeepSeek-R1
  - **Text-to-Image Generation**: Midjourney, Stable Diffusion Model, FLUX.1, Nano Banana
  - **Text-to-Video Generation**: Veo3, Kling, MiniMax



Artificial Intelligence

Machine Learning

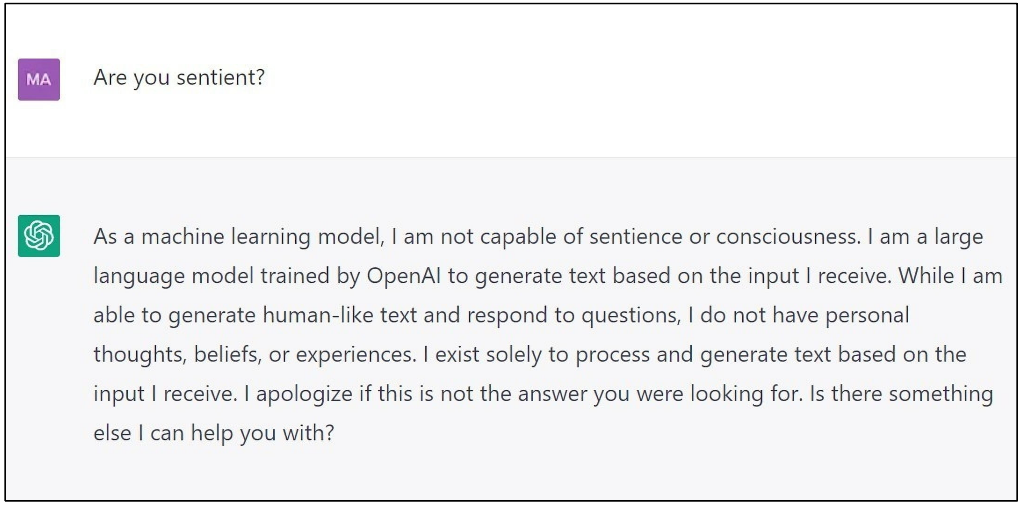
Neural Networks

Deep Learning

Generative AI

Breakthroughs in Natural Language Processing

- Large Language Models (LLMs) are a breakthrough in Natural Language Processing (NLP).
  - They use Transformers to understand and generate human language.
  - ChatGPT and similar models have achieved unprecedented levels of performance.
  - They have the potential to revolutionize human-computer interaction.

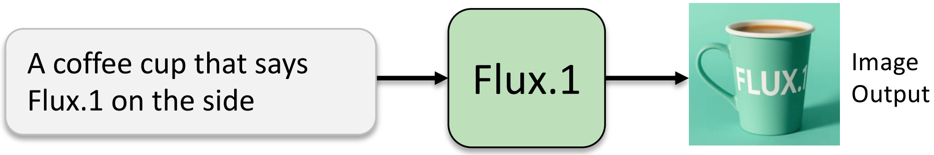


Are you sentient?

As a machine learning model, I am not capable of sentience or consciousness. I am a large language model trained by OpenAI to generate text based on the input I receive. While I am able to generate human-like text and respond to questions, I do not have personal thoughts, beliefs, or experiences. I exist solely to process and generate text based on the input I receive. I apologize if this is not the answer you were looking for. Is there something else I can help you with?

Generative AI

- **Text-to-Image Generation**
  - Midjourney, Stable Diffusion, **Flux.1**

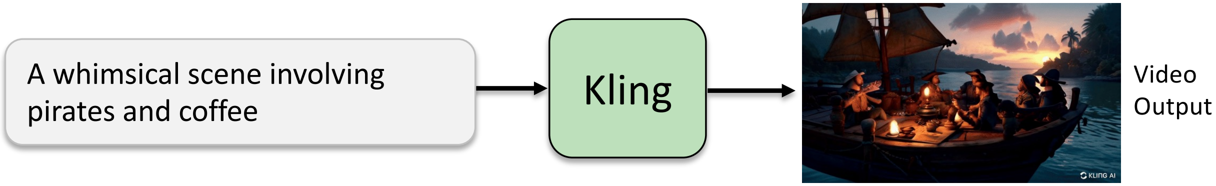


A coffee cup that says Flux.1 on the side

Flux.1

Image Output

- **Text-to-Video Generation**
  - OpenAI: Sora
  - Kuaishou: **Kling**

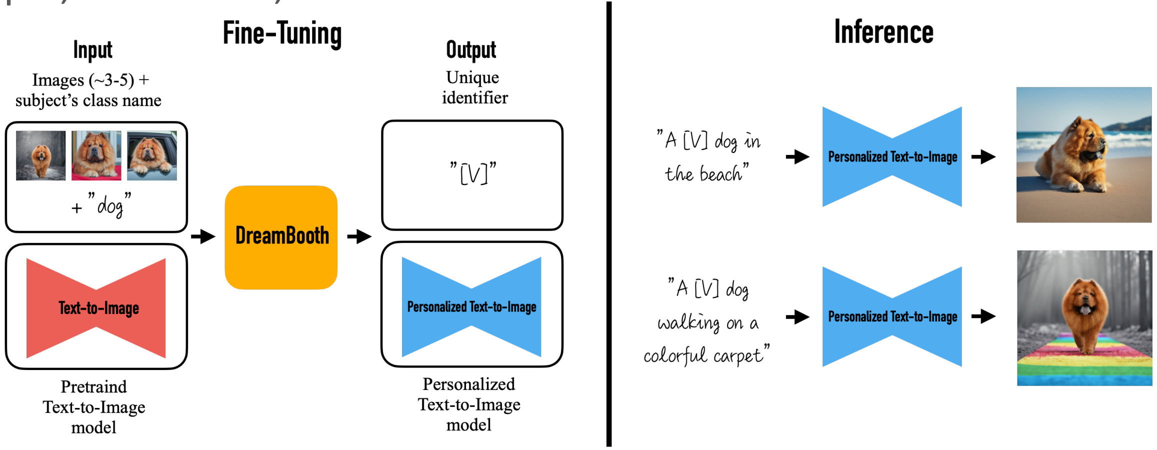


A whimsical scene involving pirates and coffee

Kling

Video Output

- In many applications, we need to **fine-tune** these models to introduce new concepts to generate desired images, concepts, characters, etc.


Fine-Tuning
Input
Images (~3-5) + subject's class name
+ "dog"
Text-to-Image
Pretraind Text-to-Image model
DreamBooth
Output
Unique identifier
"[V]"
Personalized Text-to-Image
Personalized Text-to-Image model
Inference
"A [V] dog in the beach"
Personalized Text-to-Image
"A [V] dog walking on a colorful carpet"
Personalized Text-to-Image

Google's Gemini Demo

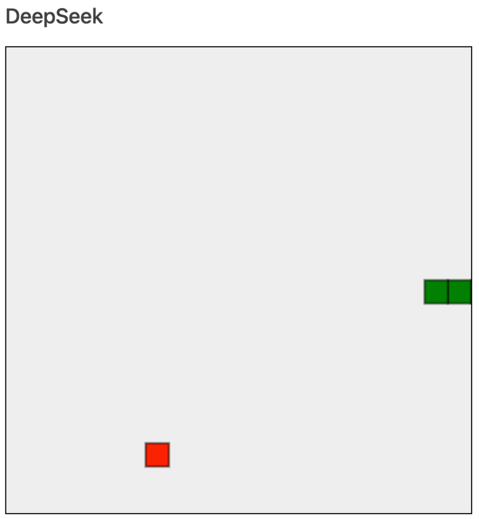
https://www.youtube.com/watch?v=UIZAiXYceBI

The Future of Coding Has Changed Indefinitely!

"Programming is not going to be essential for you to be a successful person."

Nvidia CEO Jensen Huang.

- **Just like Jensen Huang said, the LLM's latest features show us that knowing coding might not be as important as we thought.**
- Did you know that you can create a Snake or Tetris game in under 2 minutes? In this article, we'll explore DeepSeek V3, and you will understand why everyone has been talking about it these days.
- https://chat.deepseek.com/
  - Prompt: Code me an entire snake game in a single HTLM file:



EE4016
Course Content

EE4016 Course Description

- AI with Deep Learning is a hands-on course that equips aspiring AI scientists and engineers with the skills to design, train, and deploy deep neural networks using PyTorch.
- Through collaborative Python projects and real-world datasets, you’ll master key architectures—MLPs, CNNs, RNNs, Transformers and LLMs—alongside optimization, regularization, and hyperparameter tuning.
- Blending theory with practice, the course empowers you to **build scalable AI solutions and innovate at the forefront of deep learning research and application**.

Main Topics of the Course

- **Course Overview**
- **History of AI with Deep Learning**
- **Review of Linear Algebra, Calculus and Probability**
- **Multi-Layer Perceptron (MLP)**
  - Perceptron and MLP
  - Activation function, Loss function
  - Backpropagation and Gradient Descent
  - Regularizations and Optimizations
- **Convolutional Neural Networks (CNNs)**
  - Convolutional Neural Layer
  - LeNet, AlexNet, VGGNet, InceptionNet, ResNet, DenseNet, EfficientNet
  - Computer Vision Applications:
    - Object Detection, Image Segmentation
- **Recurrent Neural Network (RNN)**
  - Introduction to NLP
  - Tokenization and Word Embeddings
  - Vanilla RNN, LSTM, GRU
  - Encoder-Decoder Models and Attention
- **Transformers and Large Language Models (LLMs)**
  - Self-Attention Mechanism
  - Transformers
  - BERT, GPT, GPT-2, GPT-3, T5 and BART
  - LLM Decoding and Prompt Engineering
  - Model Quantization
  - Parameter-Efficient Fine-Tuning (PEFT)
- **Preference Alignment and Multimodality**
  - Preference Alignment: Instruction-Tuning, RLHF, DPO
  - Multimodality: CLIP, GPT-4, LLaVA

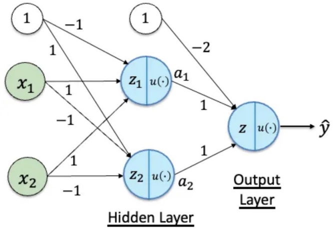
Four-Level of Understanding

1. Natural Language Understanding

- **Employ human language** to describe the concepts, problems and solutions in AI.
  - e.g. Deep learning uses multi-layered artificial neural networks to recognize complex patterns in data, resulting in state-of-the-art performance in domains such as CV and NLP.

2. Visual Understanding

- **Utilize figures** to visually represent concepts, problems and solutions in AI.



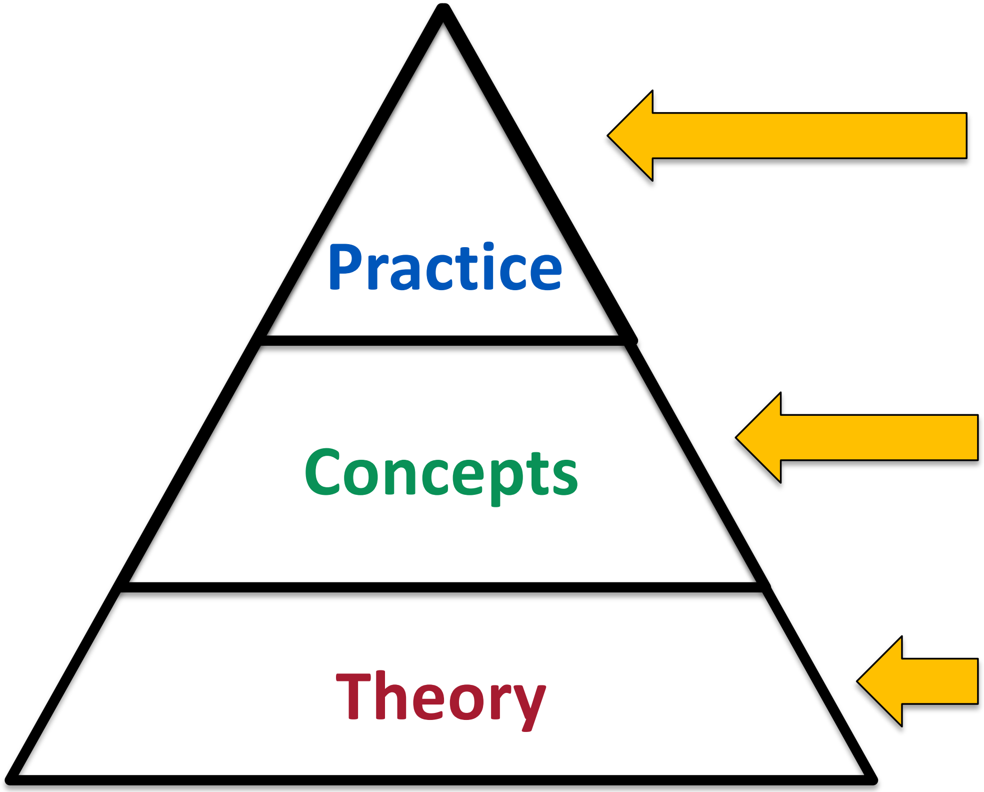
3. Mathematical Understanding

- **Utilize mathematical equations** to represent the concepts in AI.
  - e.g. A MLP model can be expressed as $P_\theta(\mathbf{x}) = \mathrm{softmax}\left(\mathbf{W}^{(2)}\mathrm{ReLU}\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\right) + \mathbf{b}^{(2)}\right)$

4. Implementation of AI System

- Implement deep learning research and applications **using Python programming language and frameworks like PyTorch.**

Learning Outcome of this EE4016 AI Course



Ability to implement theory and concepts in working code (**Python and PyTorch**)

Basic understanding of how theory is applied

Mathematics, Linear Algebra, Derivations, Deep Learning

Need to understand AI at all three levels

Python Programming Prerequisites

- Students must be familiar with Python programming.
  - The assignments and the group project will be based on Python.
  - You need to be able to read and modify the sample codes
    - Python Tutorial: https://www.youtube.com/playlist?list=PLlrxD0HtieHhS8VzuMCfQD4uJ9yne1mE6
    - OOP with Python: https://www.youtube.com/watch?v=Ej_02ICOIgs
  - You could use your only computer or **Google Colab** to run the code.
    - Google Colab Tutorial: https://www.youtube.com/watch?v=JJYZ3OE_lGo
- This course will not teach any Python package, except PyTorch.
  - We provide some background, but the class will be fast paced.
    - Deep Dive with PyTorch: The Complete Deep Learning Series
    - Zero to Mastery Learn PyTorch for Deep Learning: https://www.learnpytorch.io/
    - PyTorch for Deep Learning: https://www.youtube.com/watch?v=V_xro1bcAuA

Group Project

Learning by Doing

Group Project (20%)

- **The purpose of group project is** to creating opportunities and evaluating the skills necessary for students to independently acquire new, challenging knowledge, leveraging all available resources in their environment, and doing so in a timely fashion because promptness is of utmost importance.
- Students are required to self-organize into **five-person groups** to independently learn and execute a deep learning AI project promptly and resourcefully, with **minimal supervision**.

Stay curious, stay adaptable, stay ahead: Learn agility in the age of AI.

- Full project instructions can be found in the course project webpage:
  - https://www.ee.cityu.edu.hk/~lmpo/ee4016/projects.html

How Pick a Topic for your Group Project?

- **You pick a topic, and You pick a project type**
- Should be:
  - **Interesting**
  - **Challenging**
  - **Useful to you**

Your passion is your compass

Group Project Types

1. **Create Your Own Data** and use an existing approach ⭐
   - The main focus is on collecting data and making it useful for existing DL methods. A good example is that you collect some images and annotate them, and then train an existing image classifier to perform a specific task on these images
2. **Create Your Own Method** on an existing dataset ⭐⭐
   - Taking an existing dataset and adapting an existing method to make it your own DL method. You modify parameters, work with existing neural networks, apply what you've learned in the lecture, and aim to make them more efficient.
3. **Beat the Classics** - Implement a DL method and compare to non-DL baseline ⭐⭐⭐
   - Challenge the state-of-the-art using algorithms without DL or with DL algorithms that aim to surpass this baseline. Compare the results and demonstrate the ability to outperform the baseline, referred to as "beat”.
4. **Create Your AI-Powered Application** ⭐⭐⭐
   - Create a practical AI-powered tool or service addressing a real-world problem using advanced AI and deep learning technologies. Demonstrate creativity and technical proficiency in a functional solution.
5. **Beat the Stars** - Improve the state of the art ⭐⭐⭐⭐
   - By selecting a research paper, the goal is to demonstrate how to outperform the current state-of-the-art papers. It's important to note that due to the rapid pace of paper publication, it might be difficult to stay up-to-date with the latest advancements. Nonetheless, the objective is to select a recent paper as a baseline and attempt to beat it.

Project Team Formation (Week 3)

- Create a team of **5 people** and choose a team leader.
- The team leader is responsible for submitting the group project assessment materials, such as member list, proposal, and final report.
- Send the list of team members and the elected team leader to the Instructor at eelmpo@cityu.edu.hk
- Deadline: Jan 31, 2026

Group Project Proposal (Week 4)

- A **5-page** project proposal (not include references)
- Submit the project proposal in PDF format to CANVAS proposal assignment.
- Proposal must contain:
  - Project Title
  - Student Name, Student ID and Email Address of each member
  - Summary with goals of the project in about 300 words.
  - Other suggested content: The group proposal outlines the problem, objectives, methodology, dataset, baseline selection, experimental setup, timeline, evaluation plan, collaboration plan, risks, contingency plans, ethical considerations, and references for the group project. It serves as a roadmap, ensuring clarity and alignment among team members.
  - References
- Deadline: Feb 14, 2026 (Week 5)

Oral Presentation (Week 12 and 13)

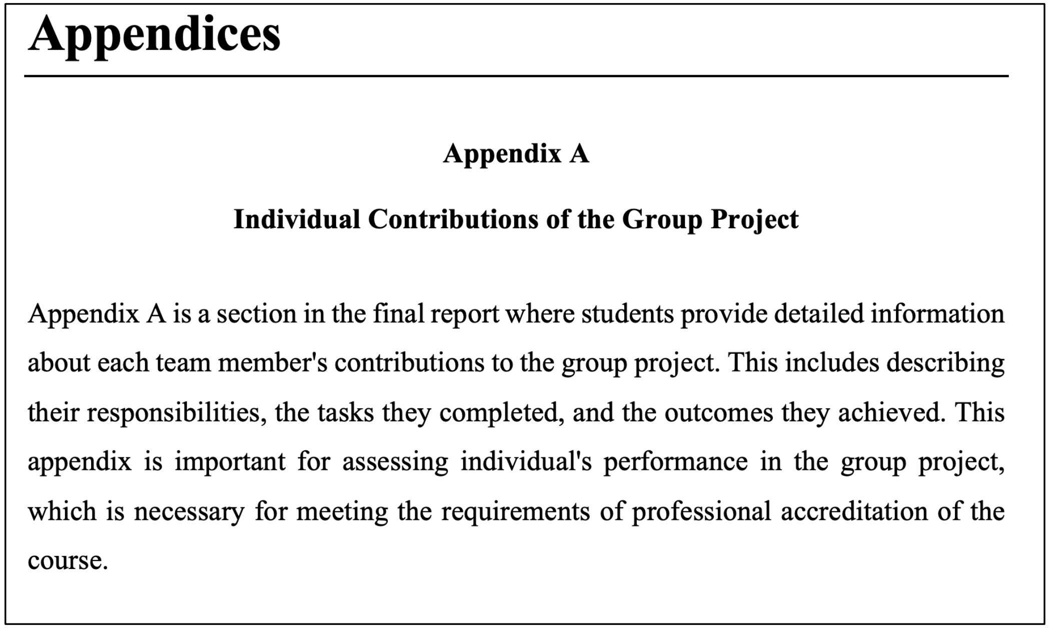
- **The oral presentation assesses students' communication skills, including their ability to clearly convey project objectives, methodology, and findings. Their response to questions gauges their understanding of the project and role within the team.**
- Every group is required to make a 10-minute Power Point presentation of the group project to the entire class on week 12 or week 13
- The presentation must include:
  - A short description of the project and its objectives
  - An explanation of the implemented algorithm and relevant theory
  - A demonstration of the working program – i.e., results obtained when running the program

Final Report, PPT, Source Code, Demo Video (Week 14)

- The final project report should be at least **25 pages in length**, including references. A final report template was provided here:
  - Final Technical Report Template: https://github.com/norlab-ulaval/template-technicalReport
  - The structure outlined in this template serves as a flexible guide rather than a rigid blueprint for the final report. While the chapters and sections presented here are commonly found in research papers or technical reports, the specific nature of the undertaken research may necessitate variations in structure. Additionally, the order of items within chapters can be adjusted accordingly. The template reflects the traditional technical report structure, which aims to demonstrate a coherent line of argument across six chapters: introduction, literature review, research design, results, discussion, and conclusions.
- Demo video is required to be a 3-4 minutes summary of the project.
- Students are also required to submit the Python source code of any implementation and PPT of the oral presentation for assessment.
- All the Final Report, PPT, Source Code and Demo video are required to submit to CANVAS Group Project Final Report
- Deadline: Apr 25, 2026 (Week 14)

Individual Assessment of the Group Project

- The final report must include an Appendix A for “Individual Contributions of the Group Project”



Appendices

Appendix A

Individual Contributions of the Group Project

Appendix A is a section in the final report where students provide detailed information about each team member's contributions to the group project. This includes describing their responsibilities, the tasks they completed, and the outcomes they achieved. This appendix is important for assessing individual's performance in the group project, which is necessary for meeting the requirements of professional accreditation of the course.

15-Minute Break

- During the break, students are highly recommended to start forming the group project team.
- More information about the Group projects will be provided and discussed in the coming few weeks.