University of Lancashire

Lecture 11

Introduction to **A**dvanced **A**rtificial **I**ntelligence

Course overview

- **Module:** Artificial Intelligence
- **Course ID**: CO3519
- **Number of Assessments:** 2
  - 1st assignment: (**Weightage**=>50% , **Contents**=> Facial Expression Recognition using ML)
  - 2nd assignment: (**Weightage**=>50% , **Contents**=> Implementation of deep learning algorithm and its visual demonstration, **Release date**: TBD)
- **Programming Language:** Python
- **Study/support Materials:** It will be provided along the lecture on the blackboard
- **Useful link for materials and lectures**
  - Blackboard

Course overview

- **Module:** Artificial Intelligence
- **Course ID**: CO3519
- **Number of Assessments:** 2
  - 1st assignment: (**Weightage**=>50% , **Contents**=> Facial Expression Recognition using ML)
  - 2nd assignment: (**Weightage**=>50% , **Contents**=> Implementation of deep learning algorithm and its visual demonstration, **Release date**: TBD)
- **Pre-requisites:** Python, Advanced programming, Basic Machine Learning
- **Study/support Materials:** It will be provided along each lecture on the blackboard.
- **Useful link for materials and lectures**
  - Blackboard

Course overview

- **Learning Objectives**
  - Introduce and familiarise you with the Advanced AI
  - Learning how perceptrons works
  - Understanding about the historical trends in AI from starting.
  - Develop, or work with, implementations of Perceptron.
- **Important Directions**

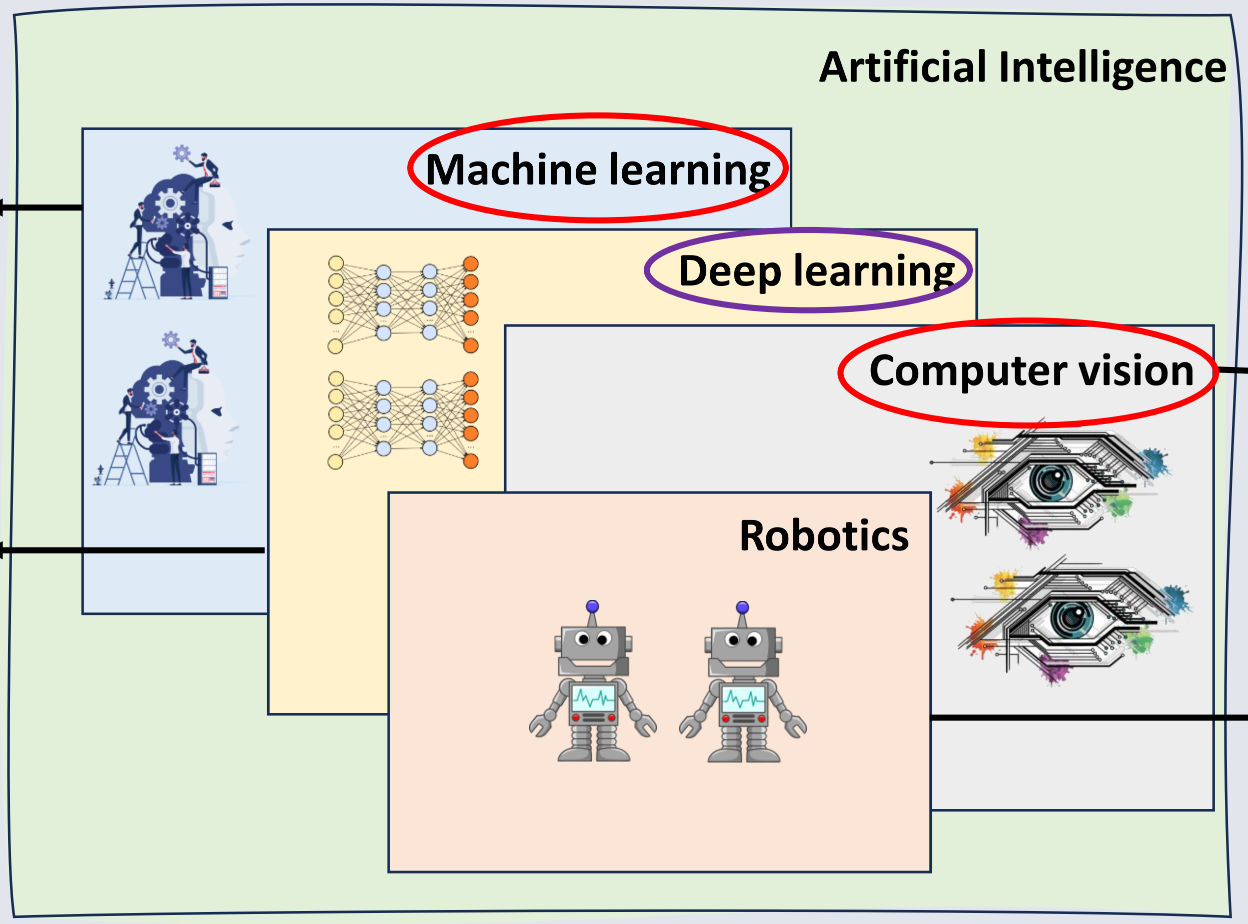
Today’s Contents

- Brief recap of traditional AI and ML concepts
- History of advanced AI (DL)
- What is Deep Learning
- What is Perceptron

Traditional AI and ML Recap

- Supervised learning
- Unsupervised learning
- Reinforcement learning

- CNN, ConvNet, ANN
- RNN, LSTM, GRU
- Autoencoder



- Image processing, recognition
- Video analysis
- Image classification

- UAV, Drones
- Automation
- Cobots

__ Covered
__ Will be covered

Traditional AI and ML Recap

Key Points Recap

AI: Simulating human intelligence (rule-based systems, expert systems).

Machine Learning: Learning patterns from data (Linear regression, Decision trees).

Limitations of Traditional ML:

Feature engineering: Handcrafting features is tedious.

Performance: Struggles with high-dimensional data like images or text.

ML Recap

Basics of Machine Learning (ML)

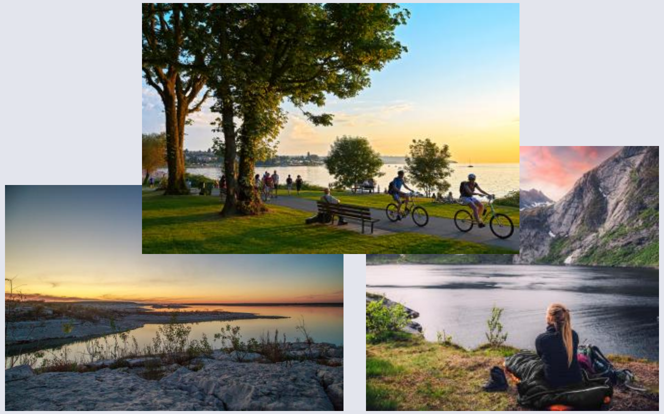
A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P**, if its performance at tasks in **T** as measured by **P**, improves with experience **E**

Machine Learning, Tom Mitchell, 1997

Example 1: Image Classification

Task: Determine if the image is indoor or outdoor
Performance measure: Probability of misclassification

Indoor



Outdoor

ML Recap

Basics of Machine Learning (ML)

Example 1: **Image classification**

A few terminologies

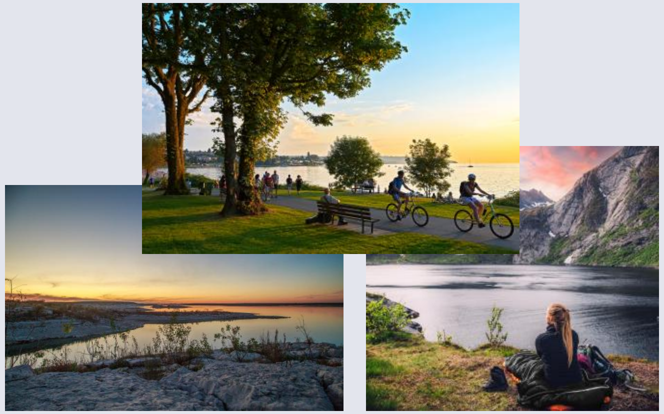
Training data: The images given for learning, **Test data:** The images to be classified

Binary classification: Classify into two classes, **Multi-class Classification:** ?

Example 1: Image Classification

Task: Determine if the image is indoor or outdoor
Performance measure: Probability of misclassification

Indoor



Outdoor

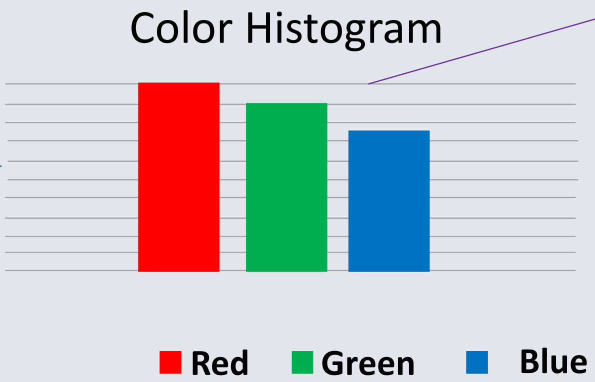
ML Recap

Basics of Machine Learning (ML)

Formulation

Extract

Features



Feature vector: x_i

Label: y_i

Indoor

0

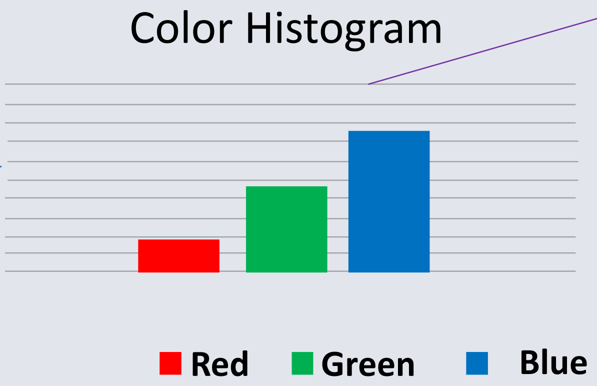
ML Recap

Basics of Machine Learning (ML)

Formulation

Extract

Features



Feature vector: xi

Label: yi

Indoor

1

Why Deep Learning?

Basic Steps in ML

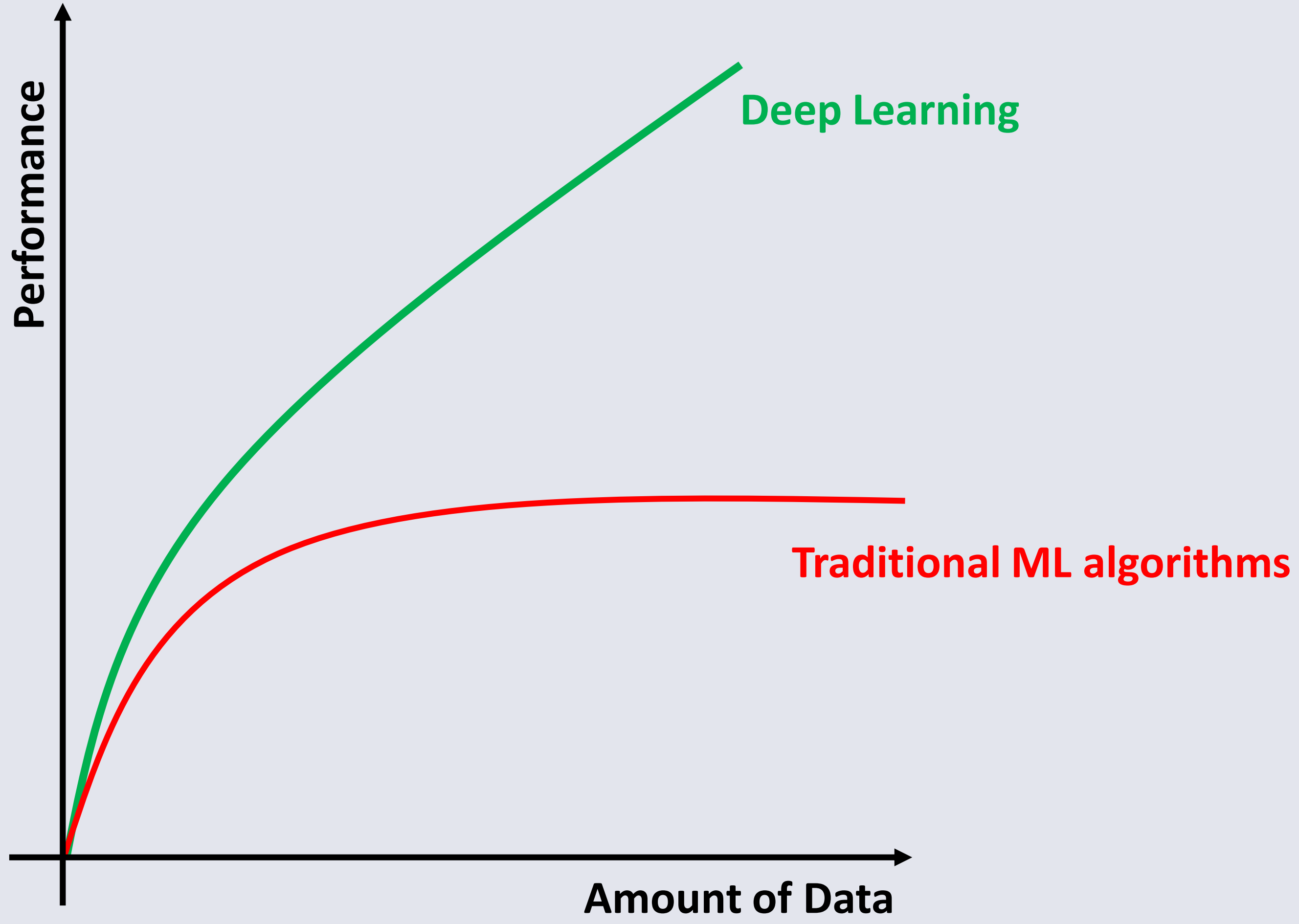
1. Collect data and extract features
2. Build model
3. Optimization: Minimize the error

DL: Requires large amounts of data

ML: Can be effective with smaller, structured datasets

DL: Can outperform machine learning for complex tasks like image recognition and natural language understanding

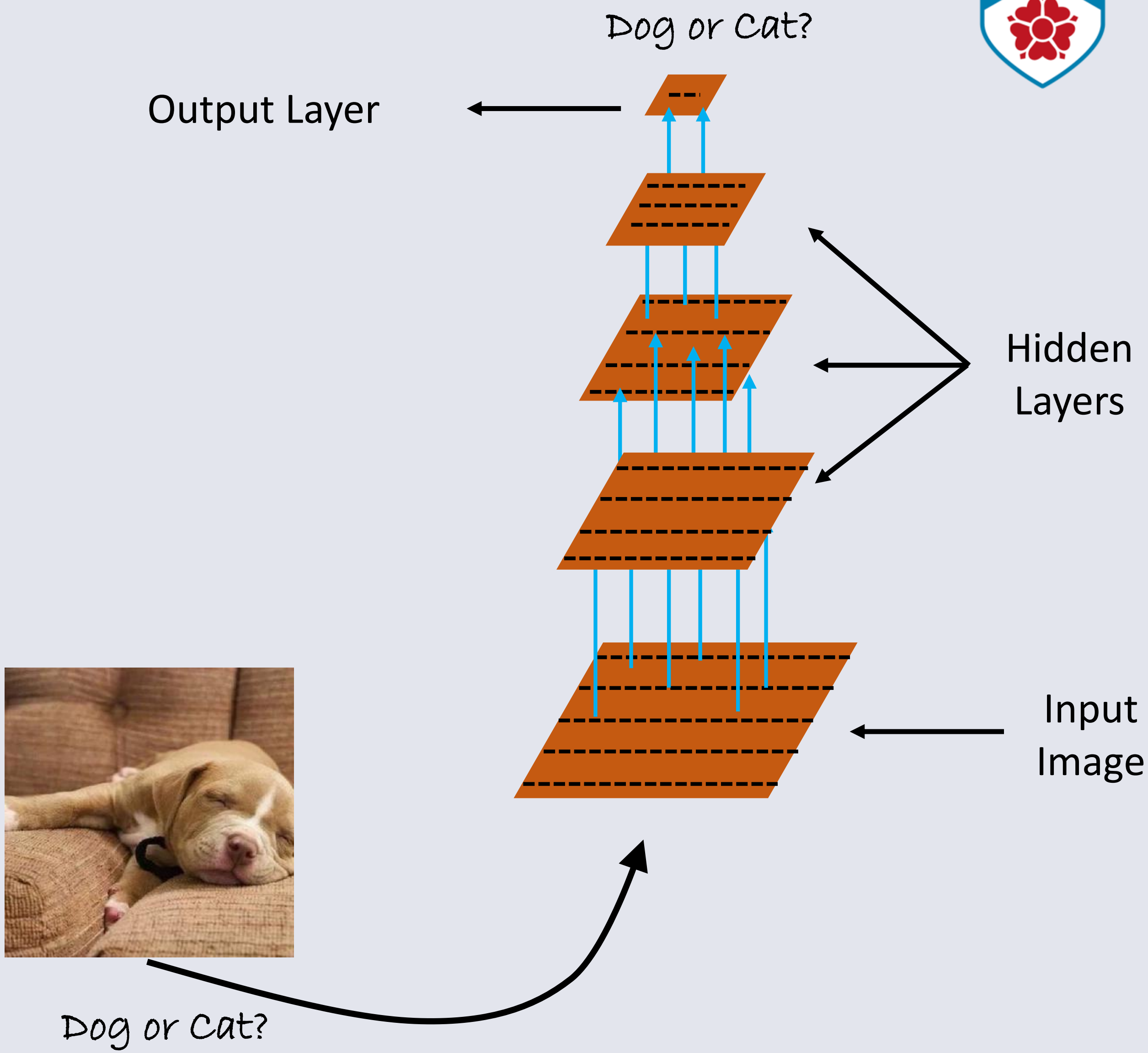
ML: Can be more efficient and easier to implement for simpler tasks



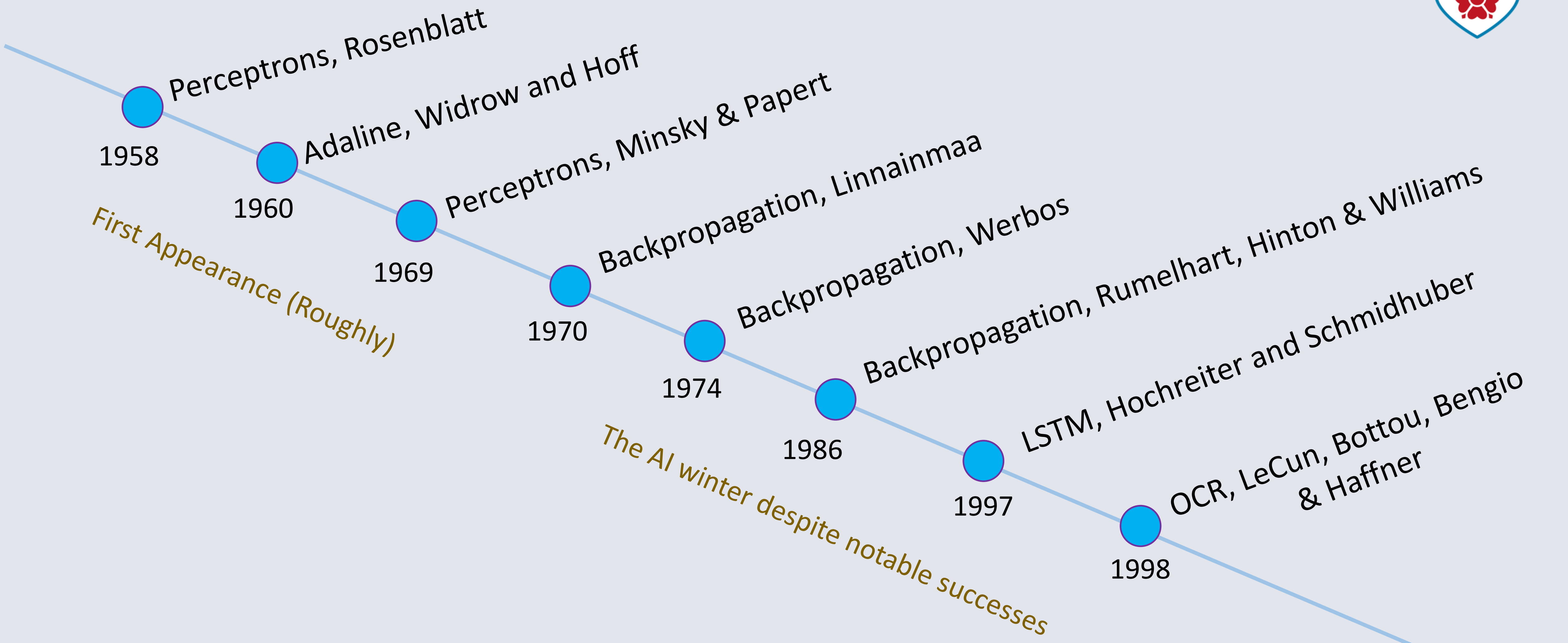
Deep Learning?

What is Deep Learning

- Sub-field of ML which attempts to learn high-level abstractions in data by utilizing hierarchical architectures.
- Using a neural network with several layers of nodes between the input and output
- The series of layers between input & output do **feature identification** and processing in a series of stages, just as our brains seem to.

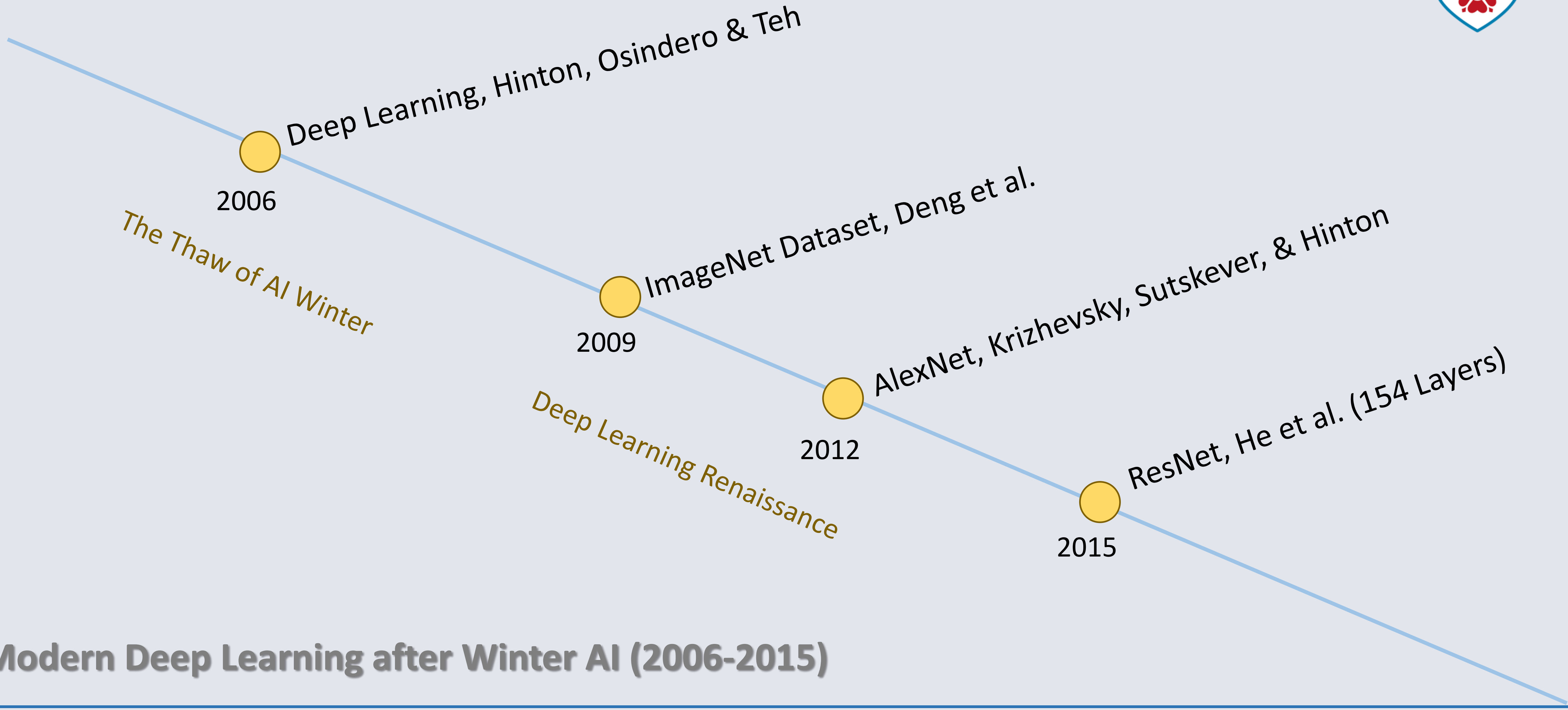


History of DL and Neural Network



Early Foundations (1958–1998)

History of DL and Neural Network



Modern Deep Learning after Winter AI (2006-2015)

History of DL and Neural Network

- 2016 — AlphaGo: DeepMind.
- 2017 — Transformer Models. Vaswani et al.
- 2018 — BERT, Google- Bidirectional Encoder Representations from Transformers
- 2019, 2020 — GPT-2 and GPT-3: OpenAI
- 2020 — AlphaFold: DeepMind

Advanced Deep Learning (2016-2020)

History of DL and Neural Network

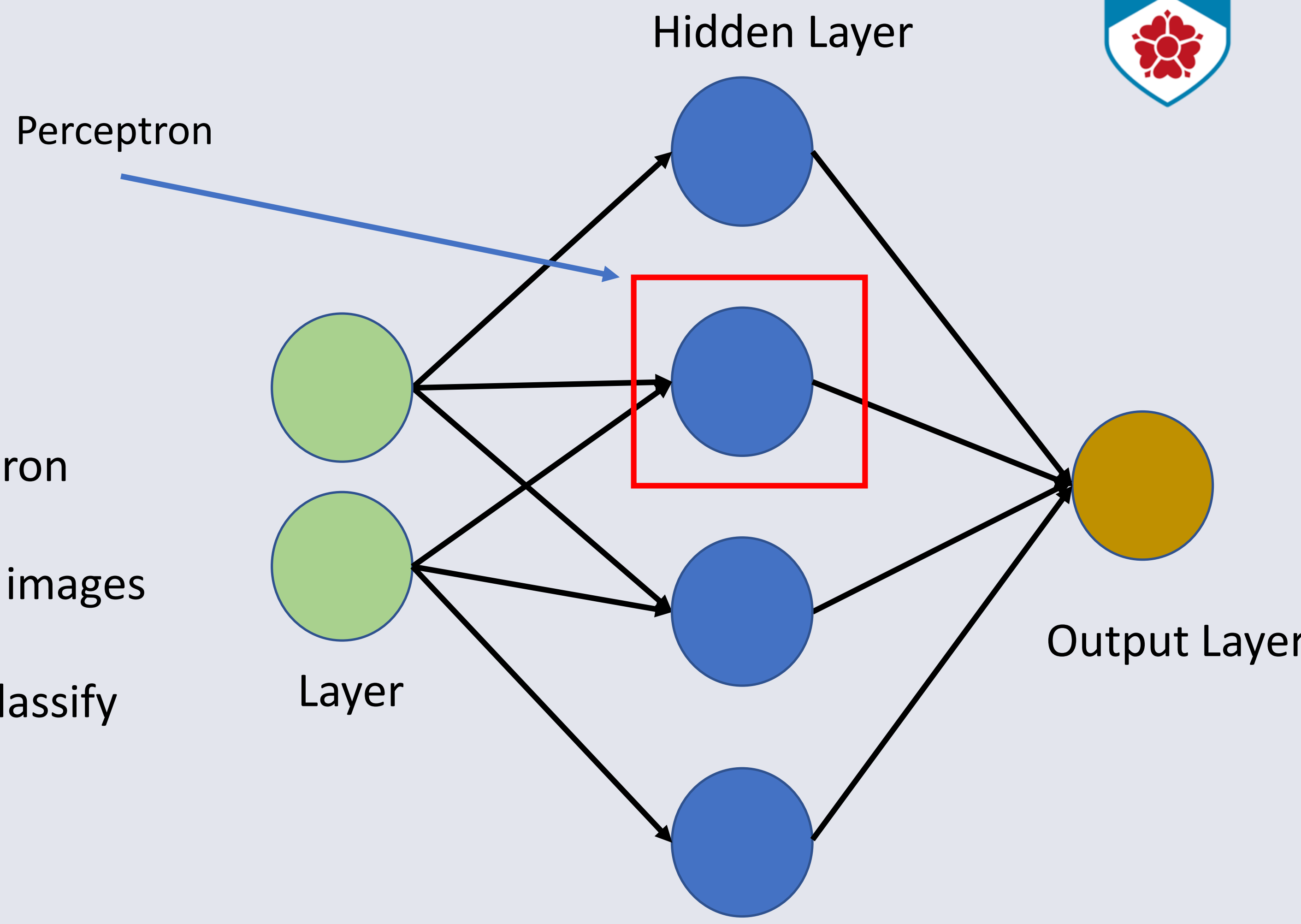
- 2021 — DALL-E: OpenAI
- 2021 — Codex: OpenAI
- 2022 — AlphaTensor: DeepMind
- 2023 — GPT-4: OpenAI
- 2023 — Large Vision Models (SAM: Segment Anything Model)
- Language + Vision Integration: CLIP (Contrastive Language-Image Pretraining)
- Generative AI Boom
- Neuromorphic Computing
- Decentralized AI

Recent Development (2021-2025)

Perceptron

- Perceptrons, Rosenblatt 1957/58

- Developed first implementation of perceptron
- Machine consist of 400 photocells, classify images
- Perceptron is one-layer network that can classify things into two parts

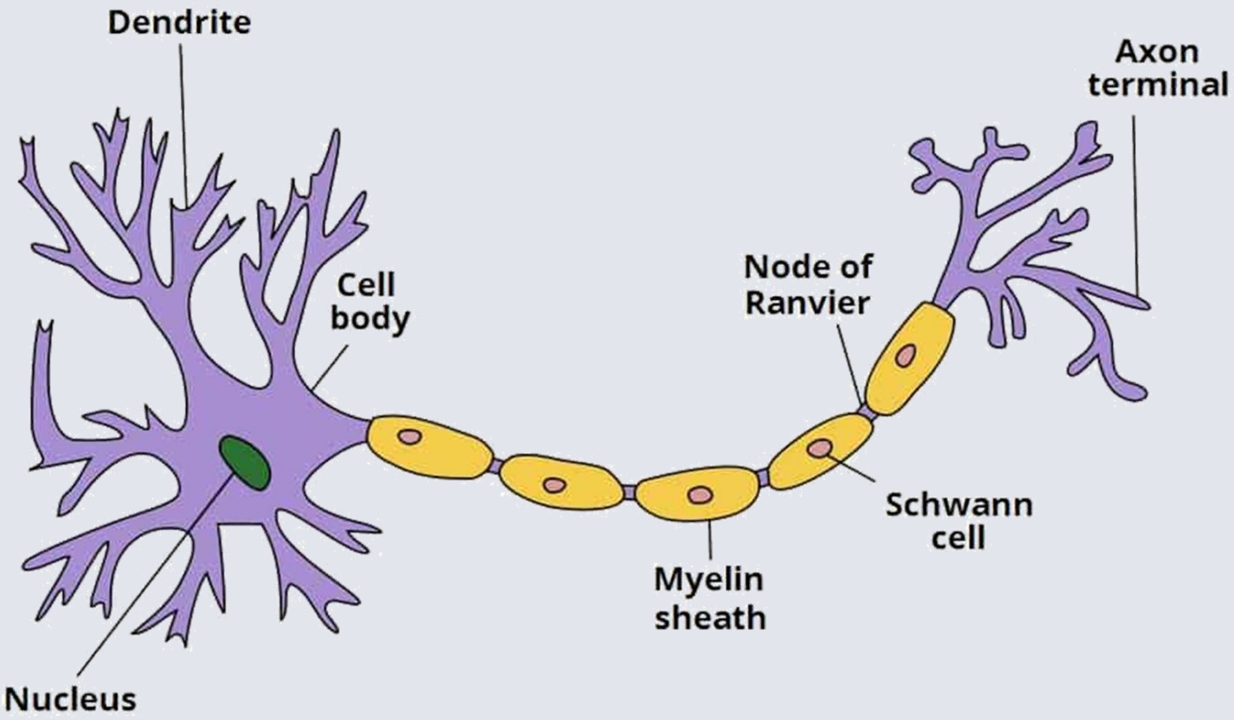


A simple Neural Network

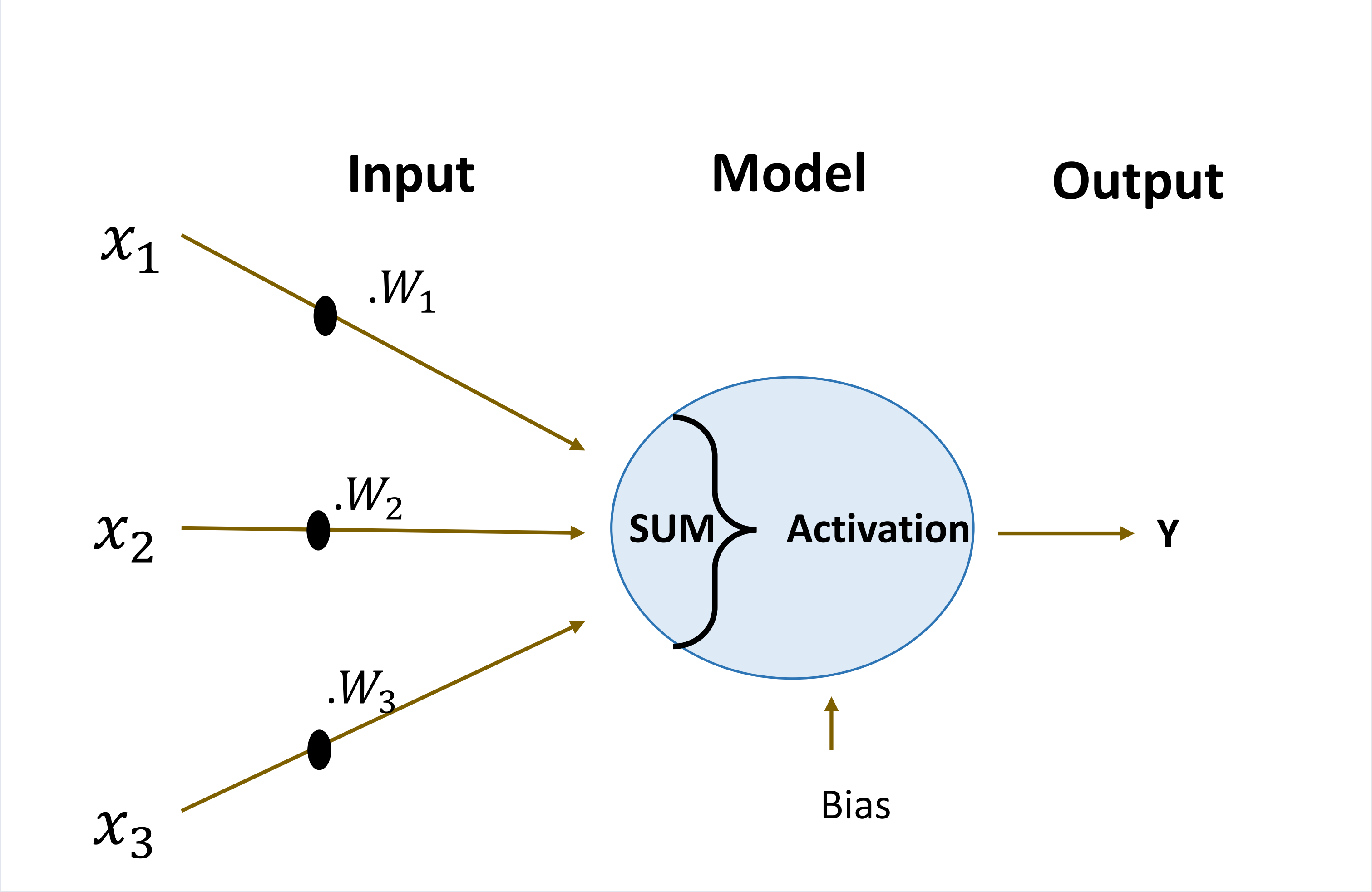
Perceptron

- Perceptrons, Rosenblatt 1957/58

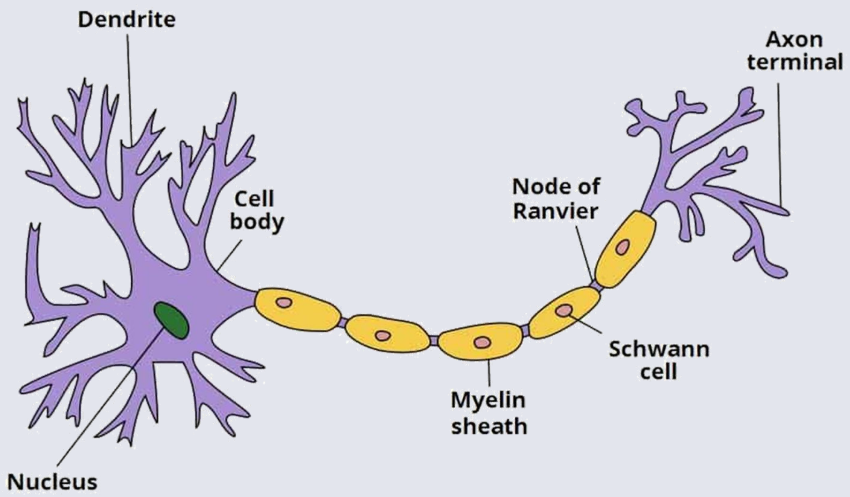
- Inspired by biological brain units called neurons.
- These neurons receive signals from other neurons.
- Depending on the strength, the neuron can either activate or off/stay.



Perceptron

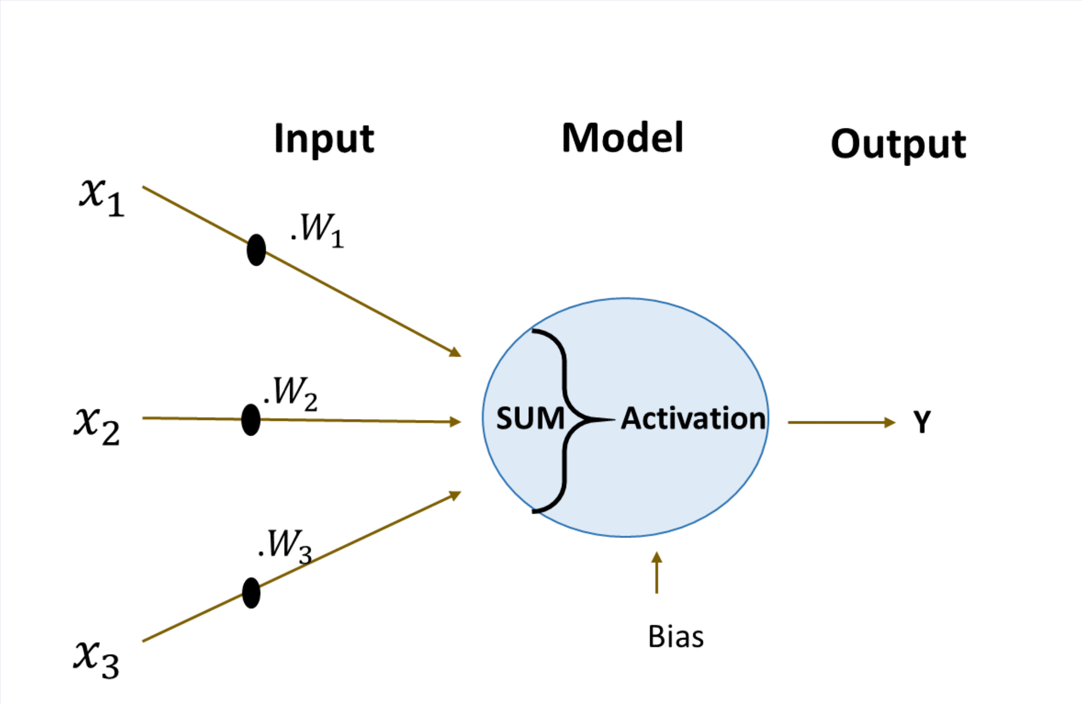


Input
Model
Output
x_1
$.W_1$
x_2
$.W_2$
x_3
$.W_3$
SUM
Activation
Y
Bias



Dendrite
Axon terminal
Cell body
Node of Ranvier
Schwann cell
Myelin sheath
Nucleus

Perceptron



$z = W_1 \times x_1 + W_2\ x_2 + W_3\ x_3 + b$

Where:
x_i: Input feature.
W_i : Weight for input xi.
b: Bias

Apply activation function to decide the output:
Y = activation (z)

In a simple perceptron, the step function is commonly used.

Activation (z) $\begin{cases} 1 \ \ if\ z\ \geq 0\ (class\ 1) \\ 0 \ \ if\ z\ < 0\ (Class\ 0) \end{cases}$

Perceptron

Example: Pass or Fail Prediction

Scenario

Predict if a student will **Pass (1)** or **Fail (0)** an exam based on:

1. Hours studied (x_1), **2.** Hours slept (x_2).

Given:

Weights: W_1 =0.6, W_2 = 0.4

Bias: b=−0.5

Inputs: x_1 = 4 (hours studied), x_2=3 (hours slept).

$z = W_1 \times x_1 + W_2\, x_2 + \mathrm{b}$

$Z = (0.6 \times 4) + (0.4 \times 3) + (-0.5)$

$Z = 2.4 + 1.2 - 0.5 = 3.1$

Activation function: Since z = 3.1 $\geq$ 0,
The output y = 1 (Pass)

```
[1] # Perceptron Function
    def perceptron(inputs, weights, bias):
        weighted_sum = sum(x * w for x, w in zip(inputs, weights)) + bias
        return 1 if weighted_sum >= 0 else 0

    # Inputs and Parameters
    inputs = [4, 3]  # Hours studied and slept
    weights = [0.6, 0.4]
    bias = -0.5

    # Prediction
    output = perceptron(inputs, weights, bias)
    print("Prediction (Pass = 1, Fail = 0):", output)
```

Perceptron

Example: Pass or Fail Prediction

Scenario

Predict if a student will **Pass (1)** or **Fail (0)** an exam based on:

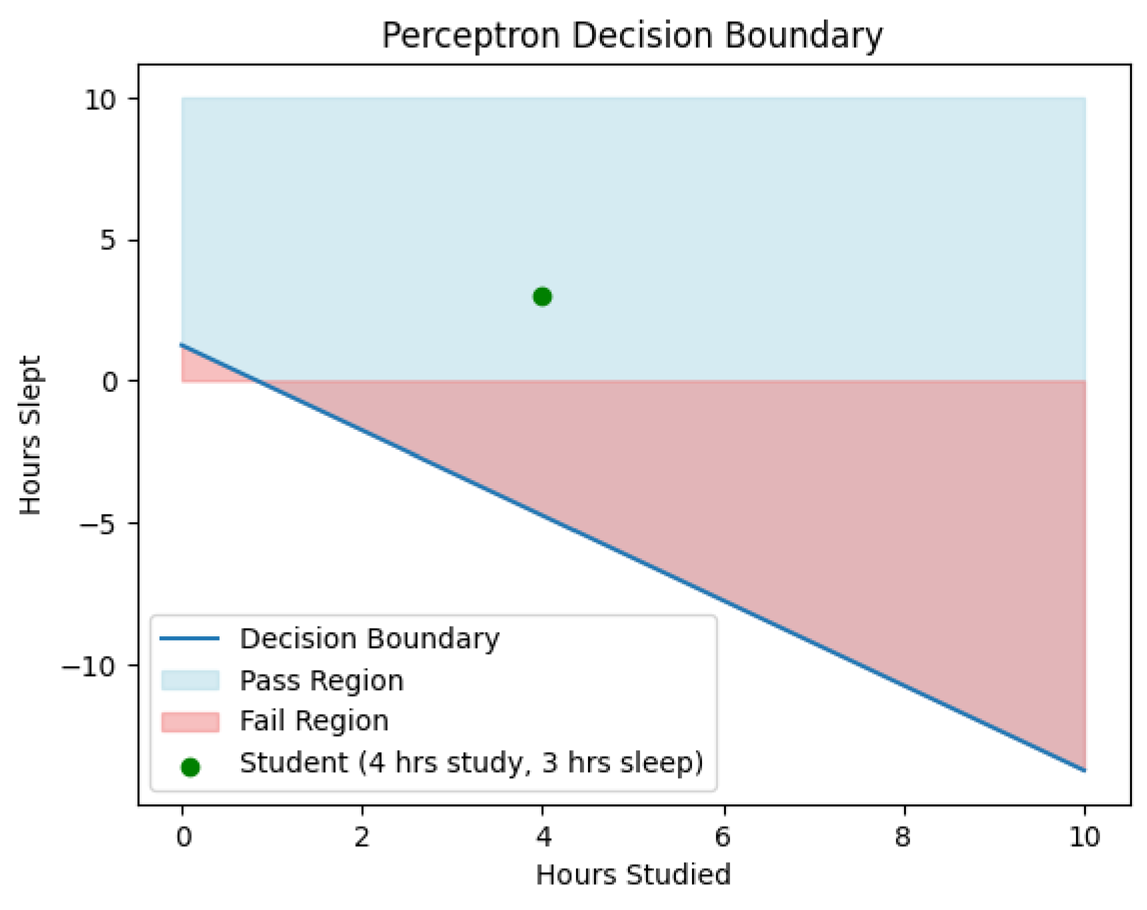
1. Hours studied (x_1), **2.** Hours slept (x_2).

Given:

Weights: W_1 =0.6, W_2 = 0.4

Bias: b=−0.5

Inputs: x_1 = 4 (hours studied), x_2=3 (hours slept).



Perceptron

Example: Spam Email Detection

```python
import numpy as np
```

Define the Activation Function. Converts the perceptron's output into a binary decision (Spam or Not Spam).

```python
def activation(x):
    return 1 if x >= 0 else 0
```

Scenario

Predict if the email is spam or not spam.

Create the Training Dataset. Provides sample email data using numerical features for training the perceptron.

```python
# Features: [Number of Suspicious Keywords, Number of Links]
X = np.array([
    [1, 0],    # Not spam
    [3, 1],    # Spam
    [0, 0],    # Not spam
    [4, 2],    # Spam
    [2, 0],    # Not spam
    [5, 3]     # Spam
])

# Labels: 1 = Spam, 0 = Not Spam
y = np.array([0, 1, 0, 1, 0, 1])

```

Next week

Convolutional Neural Networks (CNNs) – Basics

Introduction to CNNs: Why CNNs are effective for image data.

Key components: Convolution layers, pooling layers, filters, stride, padding

Activation functions, loss functions, gradient descent

Any Question?