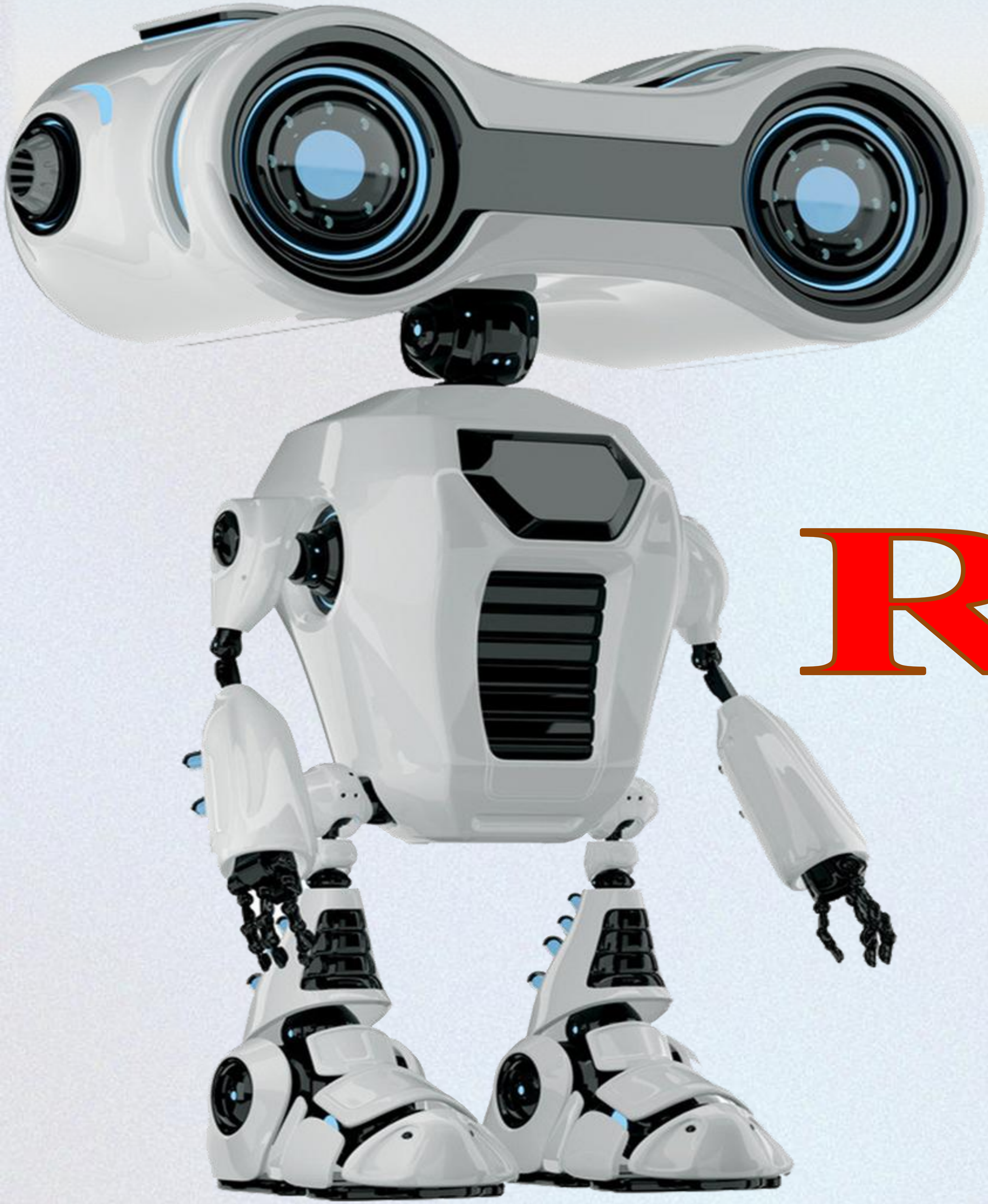
AIE111 :

Artificial Intelligence

Dr. Noha El-Sayad

Lecture 3:

Deep Learning



Revision

What is AI?

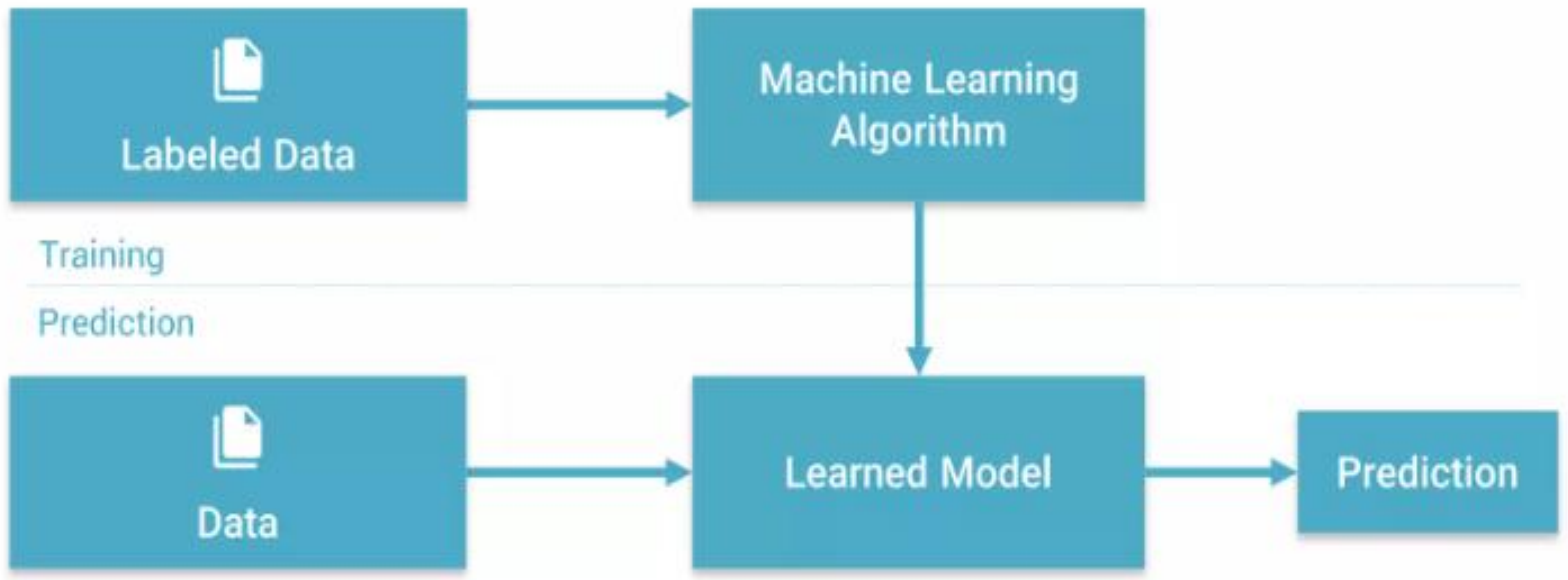
Artificial Narrow Intelligence (ANI): Machine intelligence that equals or exceeds human intelligence or efficiency at a specific task.

Artificial General Intelligence (AGI): A machine with the ability to apply intelligence to any problem, rather than just one specific problem *(human-level intelligence)*.

Artificial Superintelligence (ASI): An intellect that is much smarter than the best human brains in practically every field, including scientific creativity, general wisdom and social skills.

Machine Learning

Machine Learning is a type of Artificial Intelligence that provides computers with the ability to learn without being explicitly programmed.



Provides various techniques that can learn from and make predictions on data

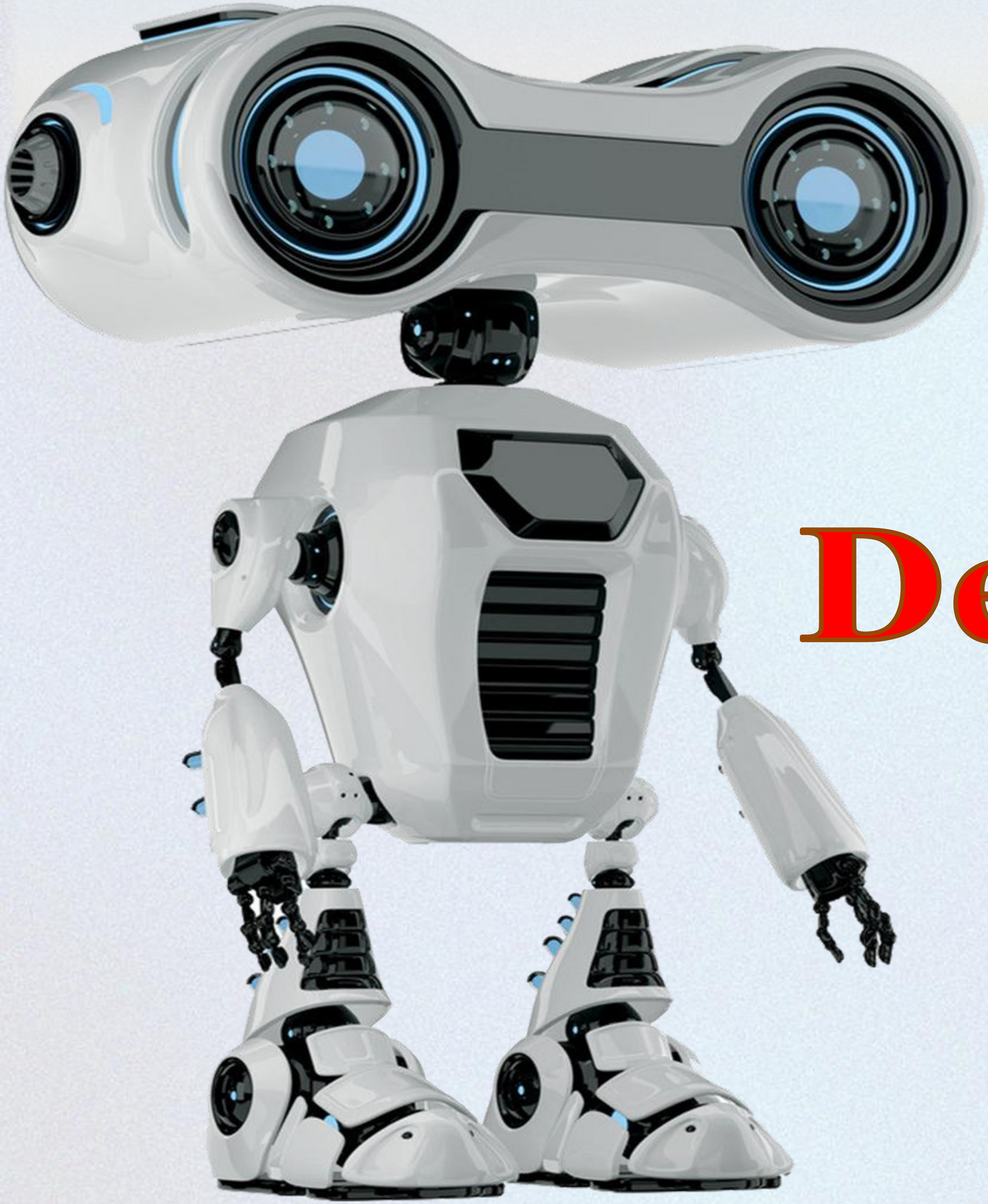
Learning Approaches

Supervised Learning: Learning with a labeled training set
Example: email spam detector with training set of already labeled emails

Unsupervised Learning: Discovering patterns in unlabeled data
Example: cluster similar documents based on the text content

Reinforcement Learning: learning based on feedback or reward
Example: learn to play chess by winning or losing

Problem Types



Deep Learning

Deep Learning

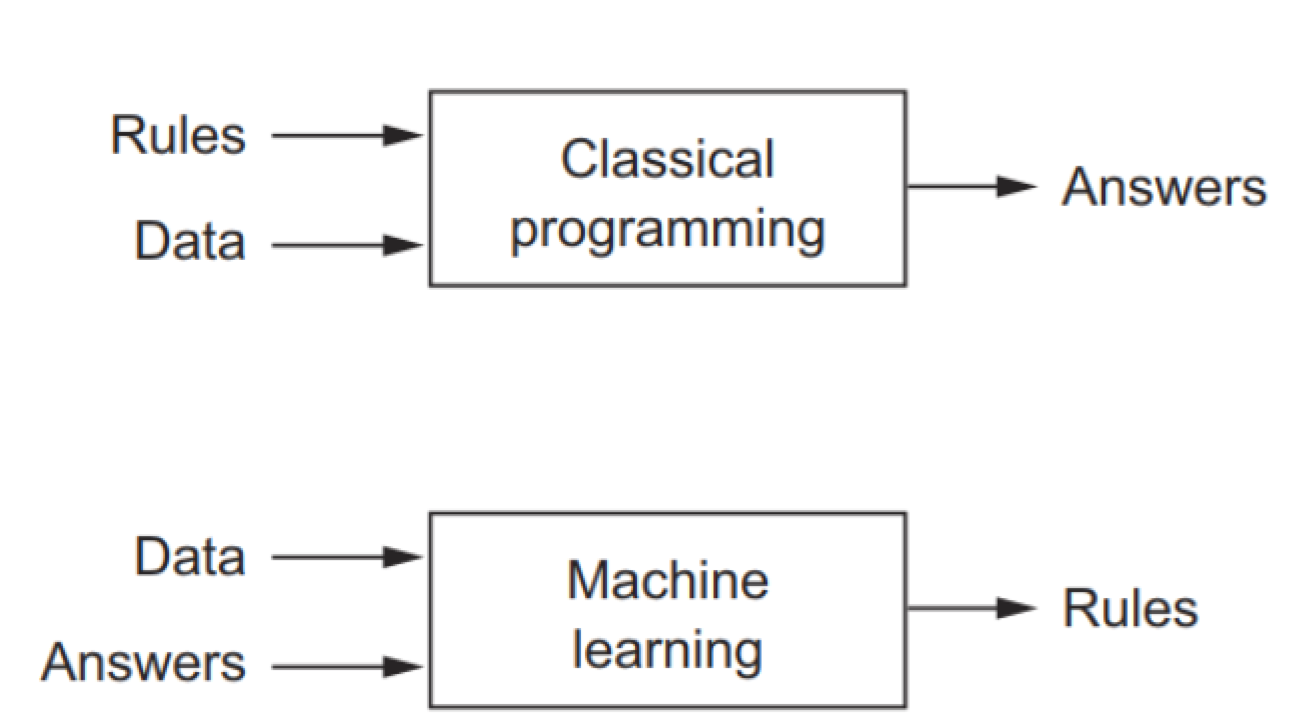
Limitations of Machine Learning

One of the big challenges with traditional machine learning models is a process called feature extraction. For complex problems such as object recognition or handwriting recognition, this is a huge challenge.

Deep Learning to the Rescue

Deep learning models are capable of focusing on the right features by themselves, requiring little guidance from the programmer.

🧠 The idea behind deep learning is to build learning algorithms that mimic the brain.



Deep Learning

Part of the machine learning field of learning representations of data. Exceptional effective at learning patterns.

Utilizes learning algorithms that derive meaning out of data by using a hierarchy of multiple layers that mimic the neural networks of our brain.

If you provide the system tons of information, it begins to understand it and respond in useful ways.

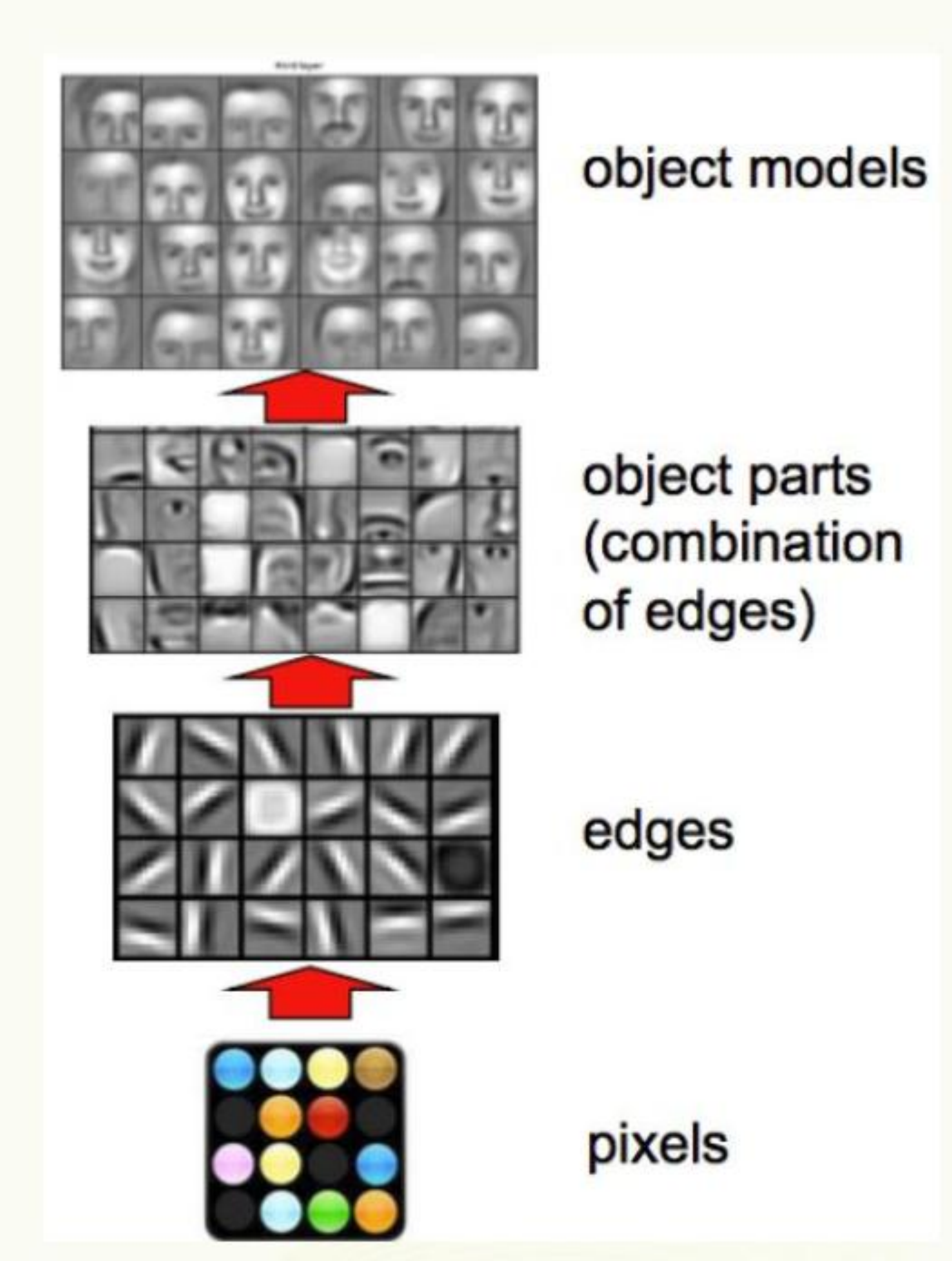
That's why it's also called:

- Layered representation learning, or Hierarchical representation learning.
- The number of layers used in the model is called the depth of the model.
- In modern deep learning, models can have dozens or even hundreds of layers to understand data more deeply and accurately.

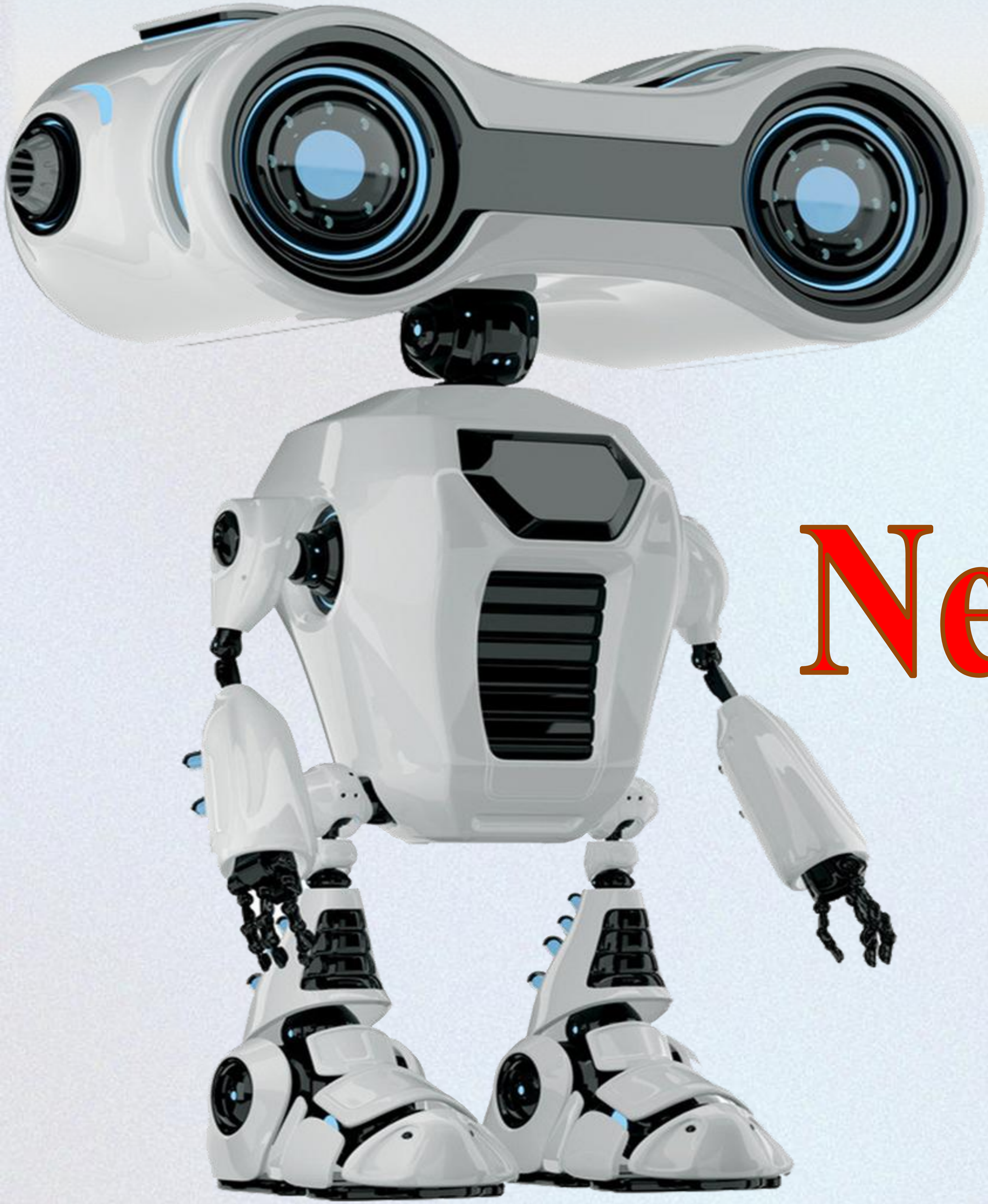
Deep Learning

Deep Learning = Machine Learning + Representation Learning

Representation Learning is a set of methods that allows a machine to be fed with raw data and to automatically discover the representations needed for detection or Classification

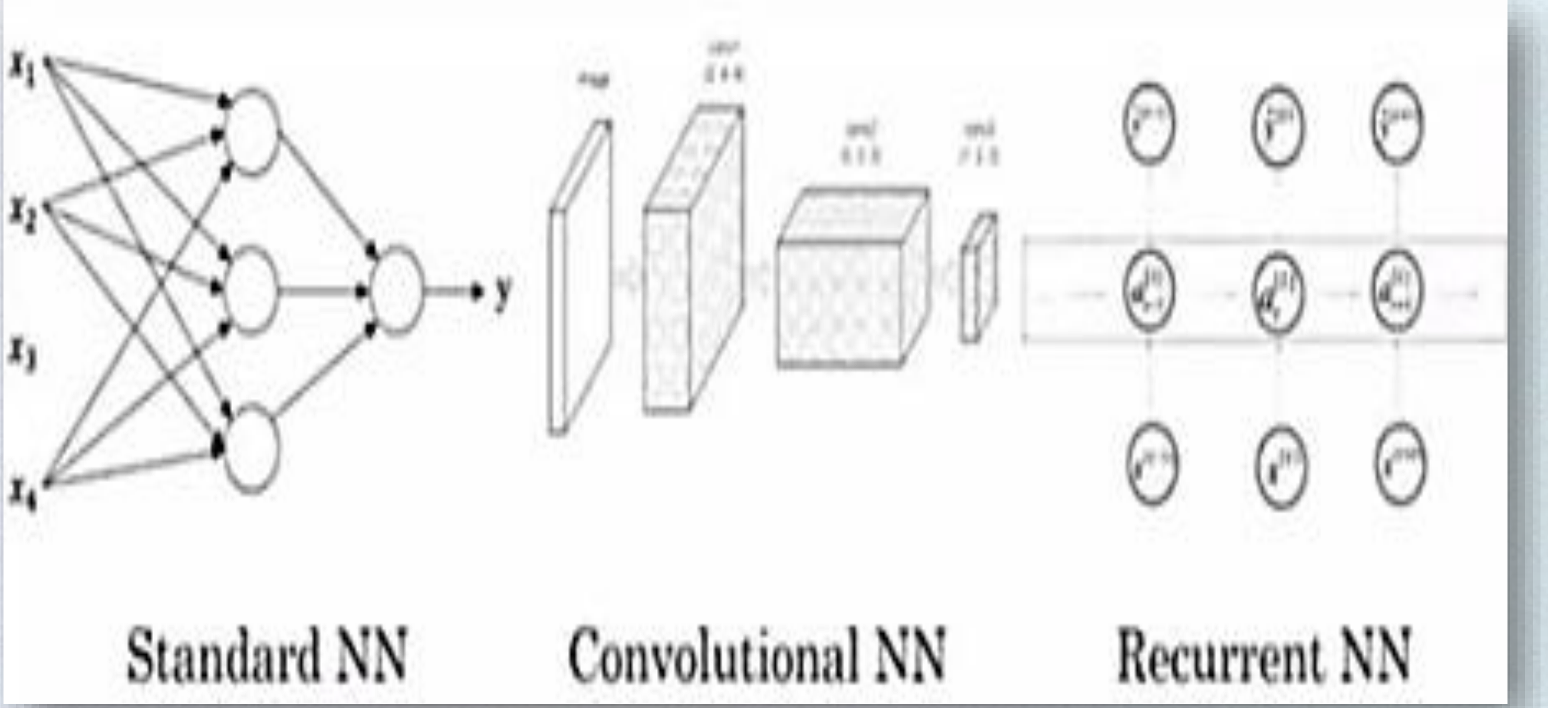


- ✓ In deep learning, we use Representation Learning models called **neural networks** to learn from data.
- ✓ These networks are made of **layers stacked on top of each other**.
- ✓ The name **"neural network"** comes from how the **human brain works**, which these models try to copy in a basic way.



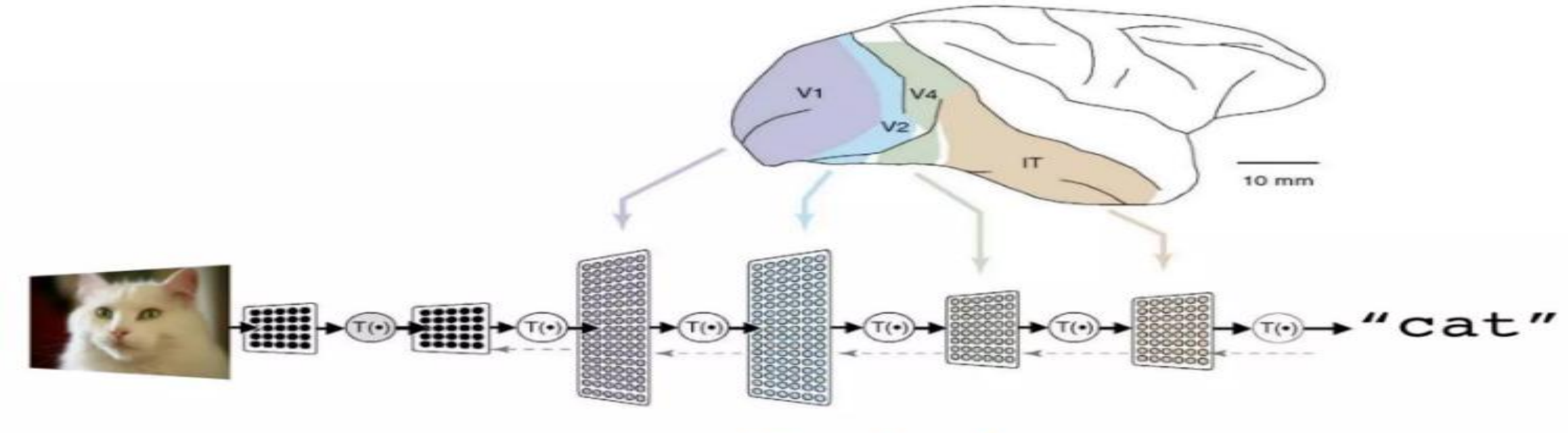
Neural Network

Different types of Neural Network in Deep Learning



An **Artificial Neural Network** (ANN) is a **general term** that refers to any computing system inspired by biological neural networks.
It includes many architectures such as:

- **Feedforward Neural Networks (FNNs)** – often what people mean by "standard"
- **Convolutional Neural Networks (CNNs)** – used in image processing
- **Recurrent Neural Networks (RNNs)** – used for sequential data
- **Deep Neural Networks (DNNs)** – simply ANNs with many hidden layers
- **Transformers, GANs, Autoencoders**, etc.

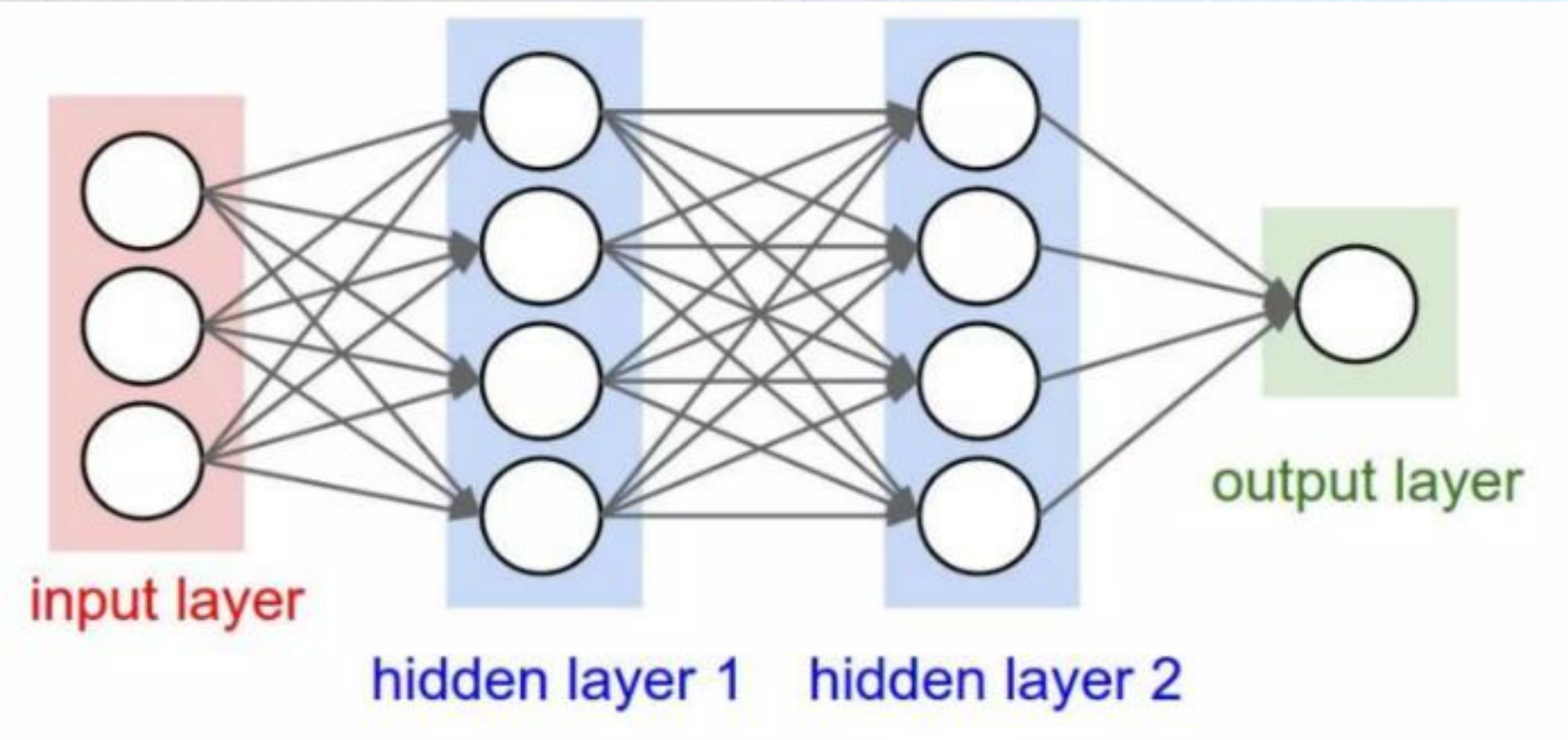


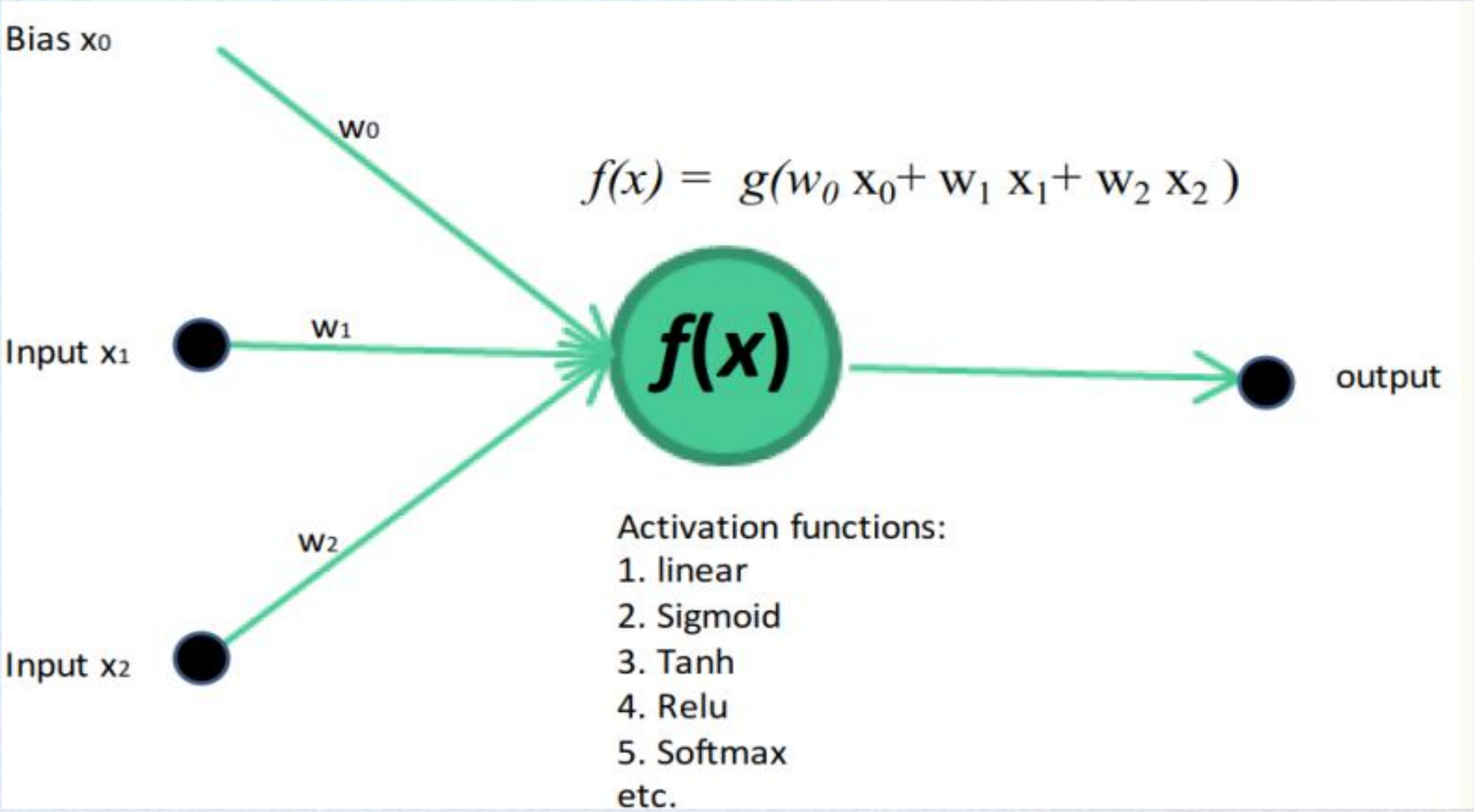
A deep neural network consists of a hierarchy of layers, whereby each layer transforms the input data into more abstract representations (e.g. edge -> nose -> face). The output layer combines those features to make predictions.

Standard / Artificial Neural Network

- ✓ **Artificial Neural Networks (ANNs)** are part of **supervised machine learning**, which means they learn from data that includes both **inputs and their correct outputs**.
- ✓ ANNs can solve both **regression** (predicting values) and **classification** (predicting categories) problems.
- ✓ They are the foundation of **deep learning**.
- ✓ A neural network learns to **recognize patterns** in the input data and then uses this knowledge to make **predictions on new data**.
- ✓ In simple terms, a neural network acts like the **brain of deep learning**, helping machines solve complex, data-driven problems by mimicking how the human brain works.

Standard / Artificial Neural Network

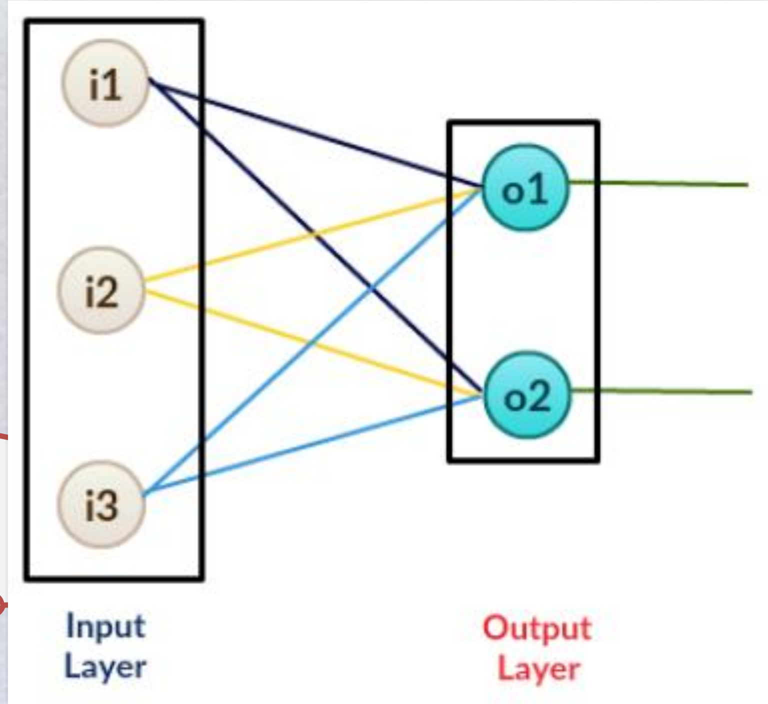




The term "**Standard Neural Network**" is not strictly defined in literature, but it's often used to refer to the **basic form of Artificial Neural Network (ANN)**—typically a **feedforward neural network** with:

- An input layer
- One or more hidden layers (fully connected)
- An output layer
- Activation functions like ReLU or sigmoid
- Trained using backpropagation and gradient descent

A perceptron is a neural network without any hidden layer.
A perceptron only has an input layer and an output layer.



Steps involved in the implementation of a neural network:

Feedforward Phase:

- We begin with a set of input features and assign initial **random weights** to the connections in the network.
- These weights are not yet optimized—they're just starting points that will be refined through training.
- The inputs are passed through the network layer by layer to compute the **predicted output**.

Backpropagation Phase:

- After obtaining the predicted output, we compute the **error** by comparing it to the actual (target) output.
- This error is then propagated backward through the network using an optimization algorithm—typically **gradient descent**.
- During this process, the weights are **adjusted** to minimize the error, improving the network's performance over time.

Steps involved in the implementation of a neural network:

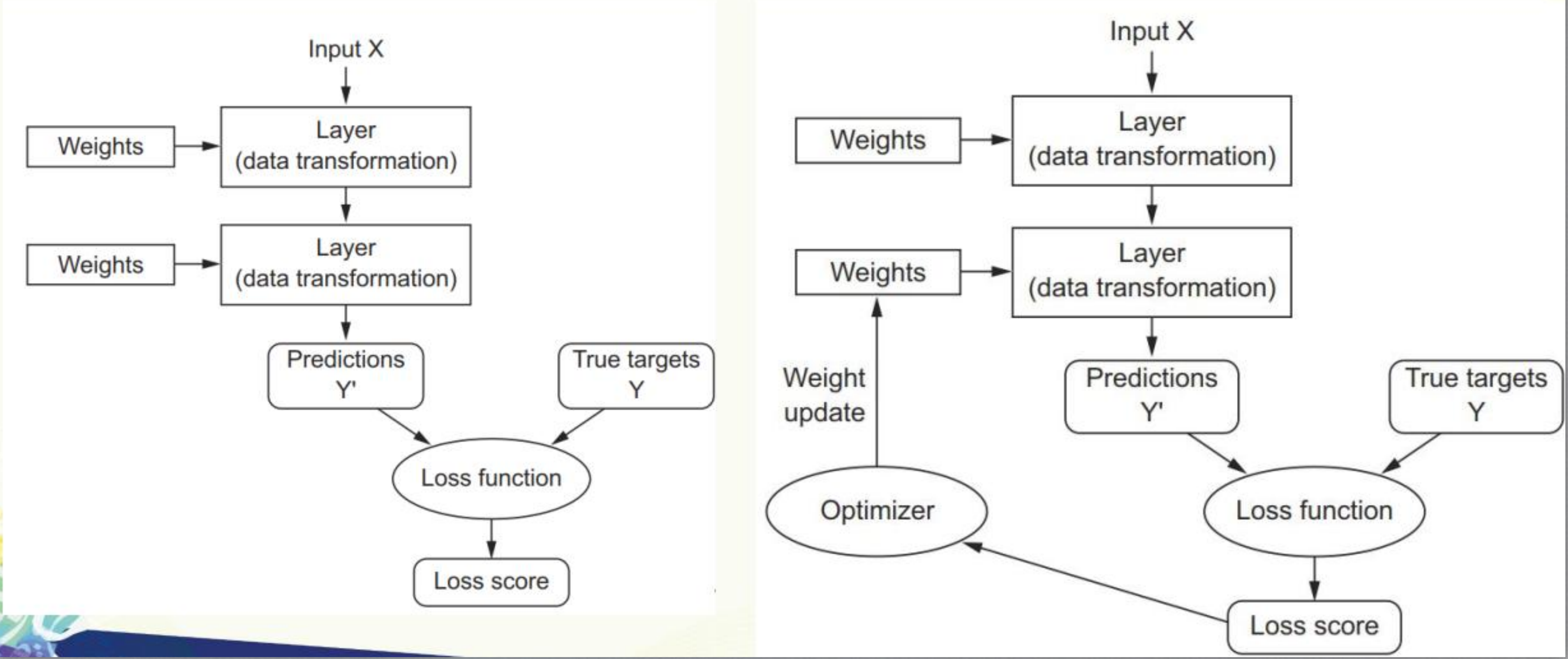
Feedforward Phase:

- We begin with a set of input features and assign initial **random weights** to the connections in the network.
- These weights are not yet optimized—they're just starting points that will be refined through training.
- The inputs are passed through the network layer by layer to compute the **predicted output**.

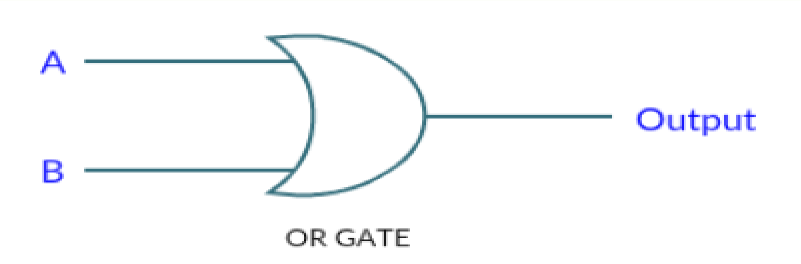
Backpropagation Phase:

- After obtaining the predicted output, we compute the **error** by comparing it to the actual (target) output.
- This error is then propagated backward through the network using an optimization algorithm—typically **gradient descent**.
- During this process, the weights are **adjusted** to minimize the error, improving the network's performance over time.

Flow chart for a simple NN



Example (logical OR Gate)



Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

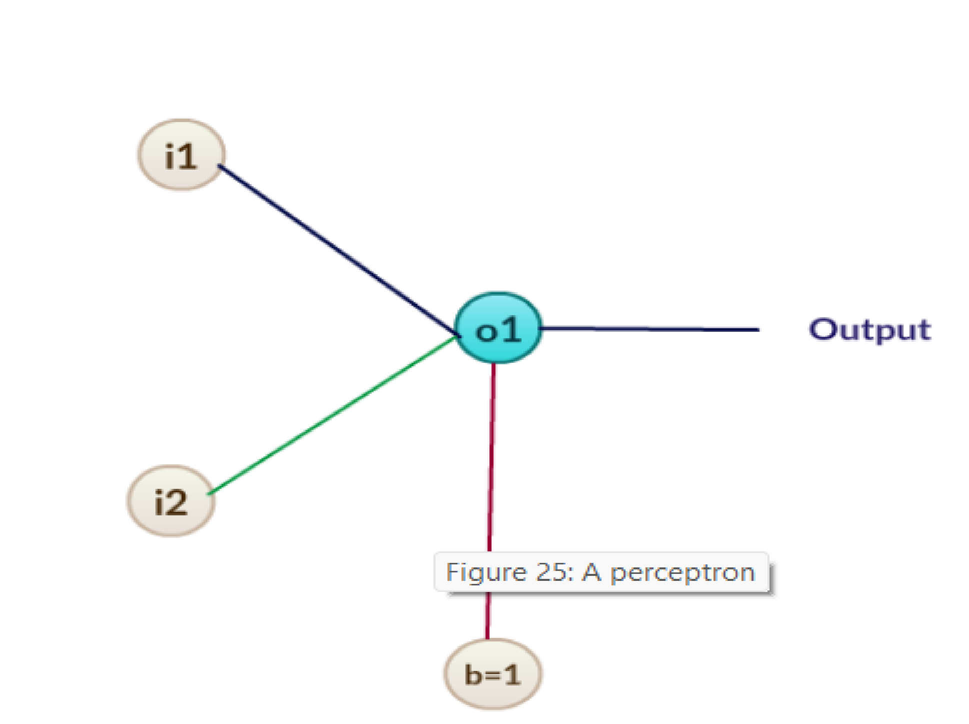


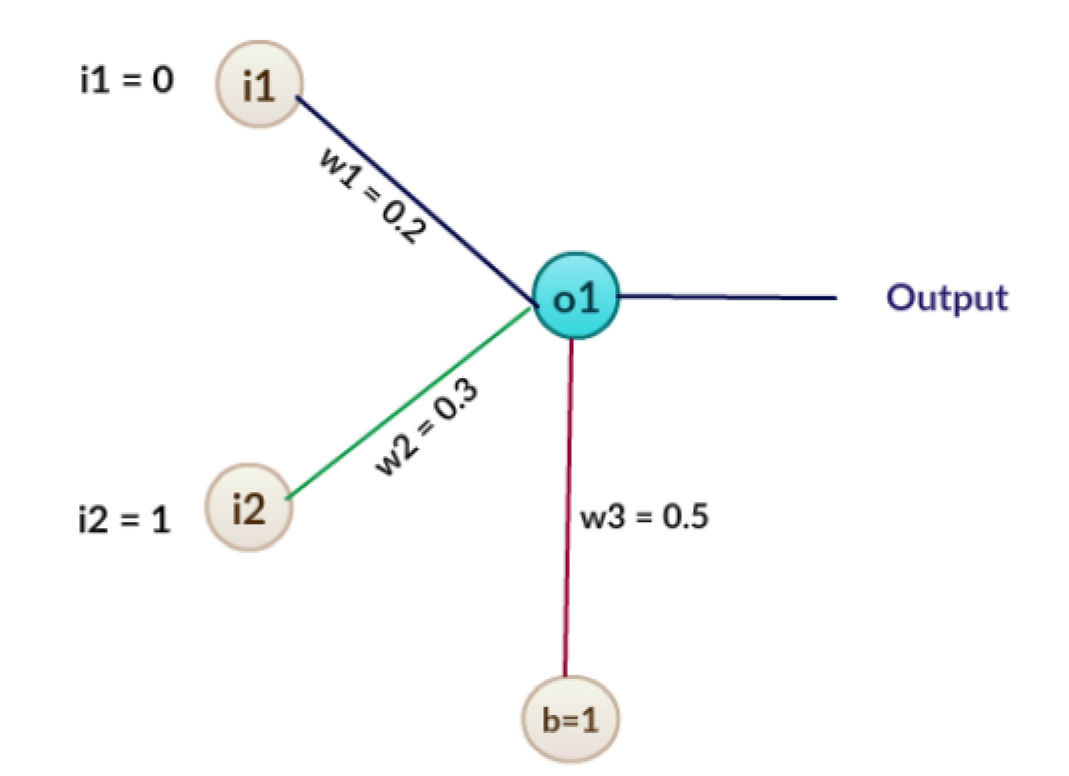
Figure 25: A perceptron

Input for o1 = w1*x1 + w2*x2 + b*x3

= 0.2*0 + 0.3*1 + 0.5*1

= 0 + 0.3 + 0.5

= 0.8



$$f(X) = \frac{1}{1 + e^{-X}}$$

Output for 01 = $\frac{1}{1+e^{-0.8}}$

= 0.68997

$$MSE = \sum \frac{1}{2} * (target - output)^2$$

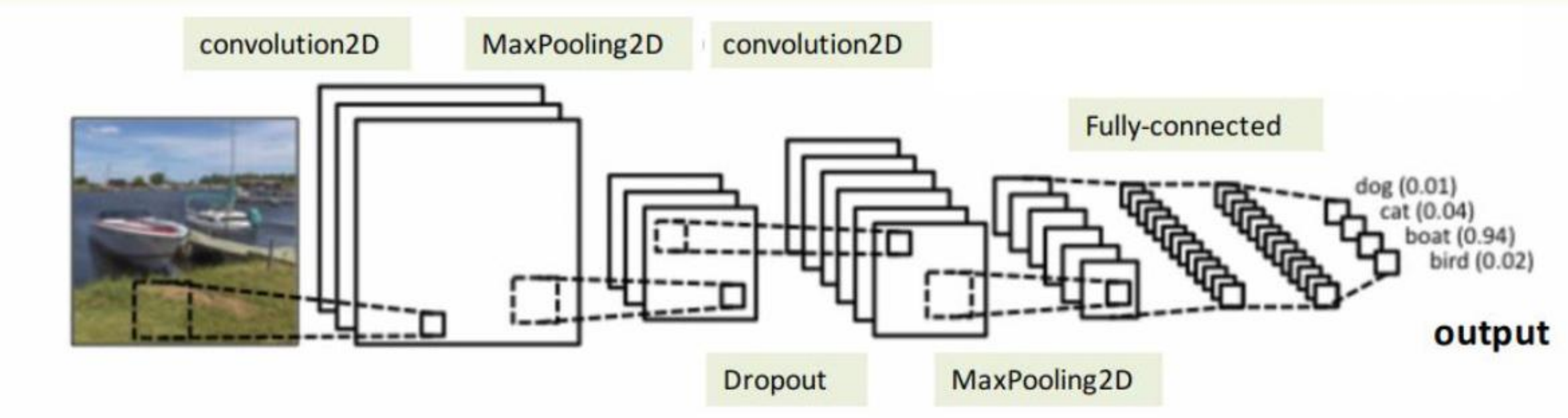
$$MSE = \frac{1}{2} * (1 - 0.68997)^2 = 0.048059$$

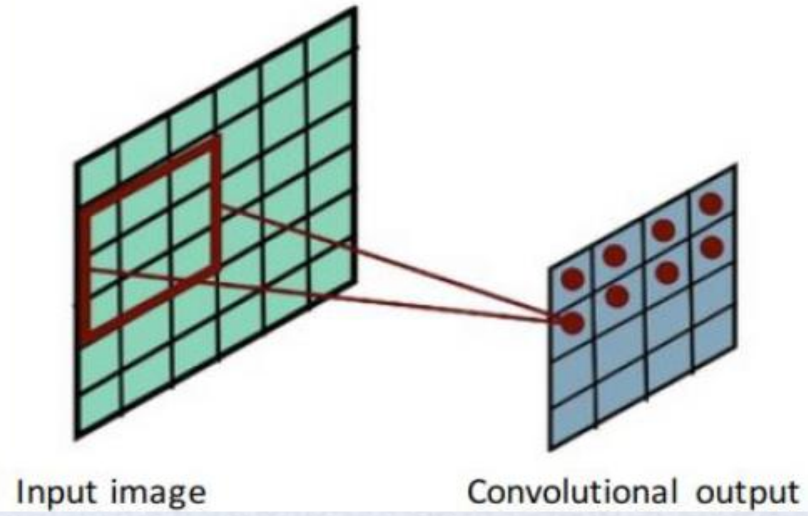
Convolution Neural Network

A **Convolutional Neural Network** is a specialized type of Artificial Neural Network designed primarily for **image processing and pattern recognition** tasks.
Unlike basic feedforward networks, CNNs are able to **automatically detect spatial hierarchies of features** (like edges, shapes, textures) in images.

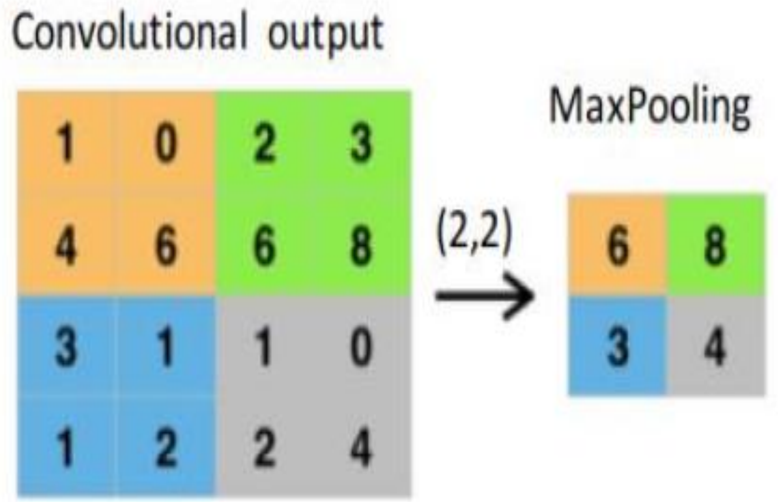
Main Components of a CNN:

1. **Input Layer:**
   - Accepts image data, usually as a 2D or 3D array (e.g., height × width × channels).
2. **Convolutional Layer:**
   - Applies **filters (kernels)** that slide over the image to extract features such as edges or textures.
   - Produces **feature maps** that highlight the presence of specific patterns.
3. **Activation Function (e.g., ReLU):**
   - Introduces **non-linearity** to help the network learn complex patterns.
4. **Pooling Layer (e.g., Max Pooling):**
   - **Reduces the spatial size** of the feature maps.
   - Helps in **dimensionality reduction**, making computation faster and reducing overfitting.
5. **Fully Connected Layer (Dense Layer):**
   - Acts as a classifier by connecting the learned features to the output layer (e.g., class probabilities).
6. **Output Layer:**
   - Produces the final prediction (e.g., the class of an image).

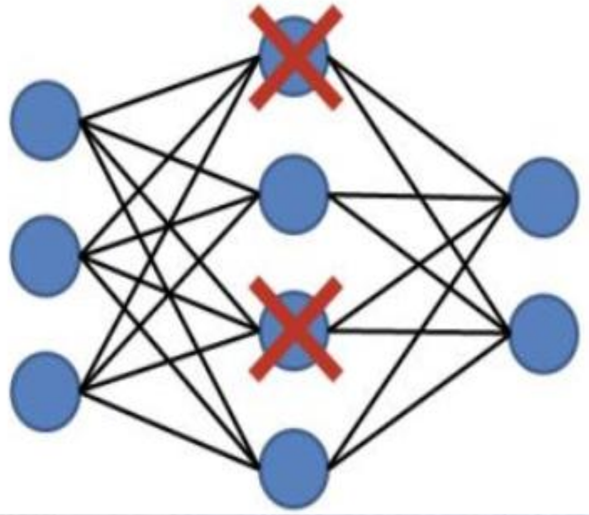




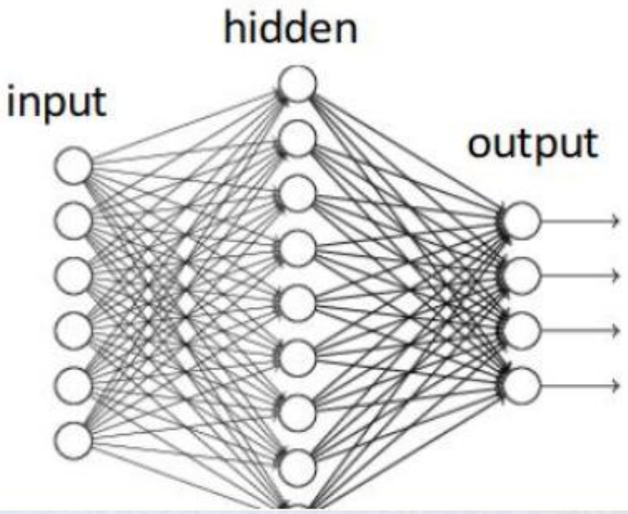
Convolutional Layer: filters work on every part of the image, therefore, they are searching for the same feature everywhere in the image.



MaxPooling: usually present after the convolutional layer. It provides a down-sampling of the convolutional output



Dropout: randomly drop units along with their connections during training. It helps to learn more robust features by reducing complex co-adaptations of units and alleviate overfitting issue as well.



Fully-connected layer (dense): each node is fully connected to all input nodes, each node computes weighted sum of all input nodes. It has one-dimensional structure. It helps to classify input pattern with high-level features extracted by previous layers.

Convolution Neural Network

🧠 **CNN Main Layers:**

1. **Convolution (Filter):**

   Detects features (edges, shapes) by sliding small filters over the image.

2. **Max Pooling:**

   Reduces image size by keeping the most important information.

3. **Dropout:**

   Randomly disables some neurons during training to prevent overfitting.

4. **Fully Connected:**

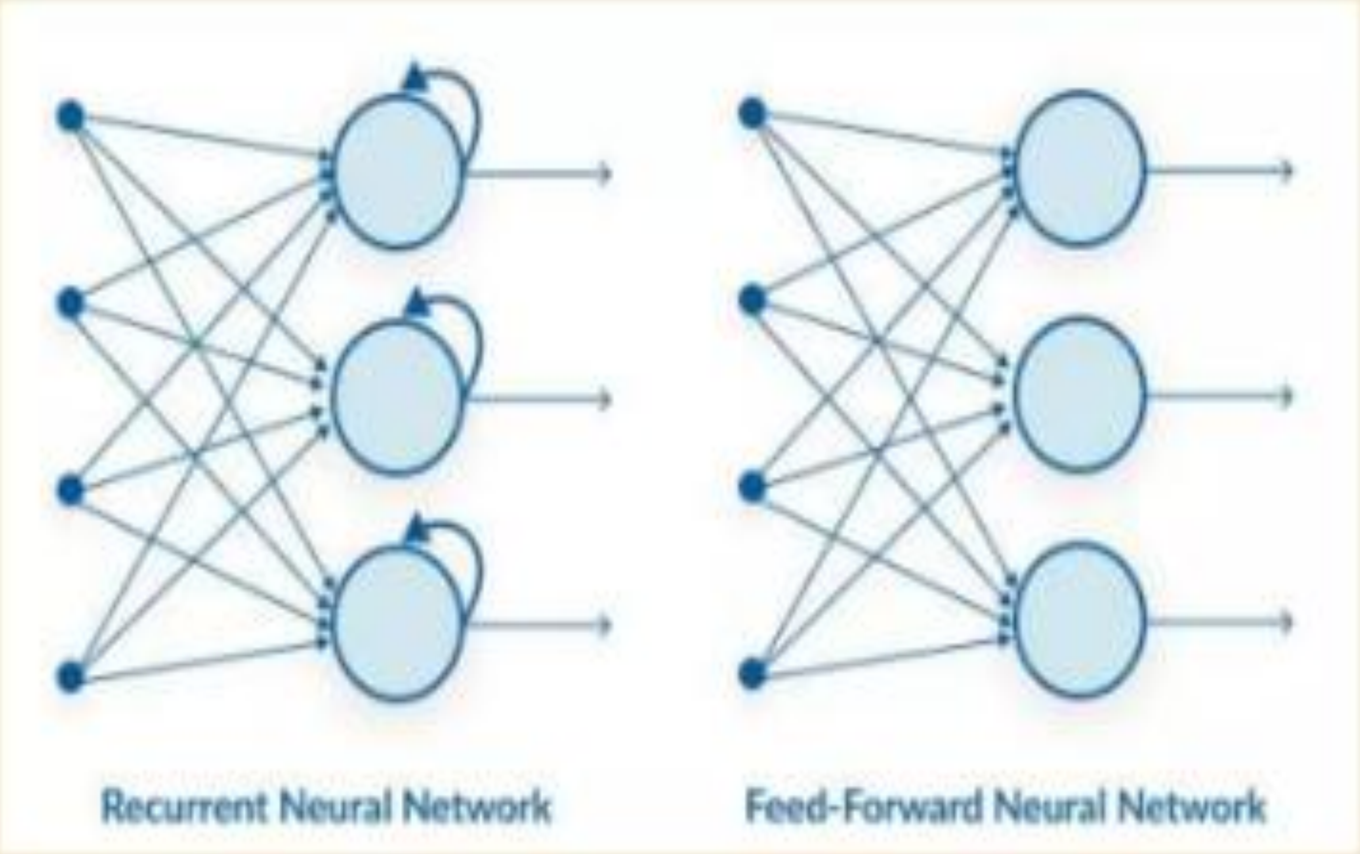
   Flattens features and makes the final prediction (e.g., image class).

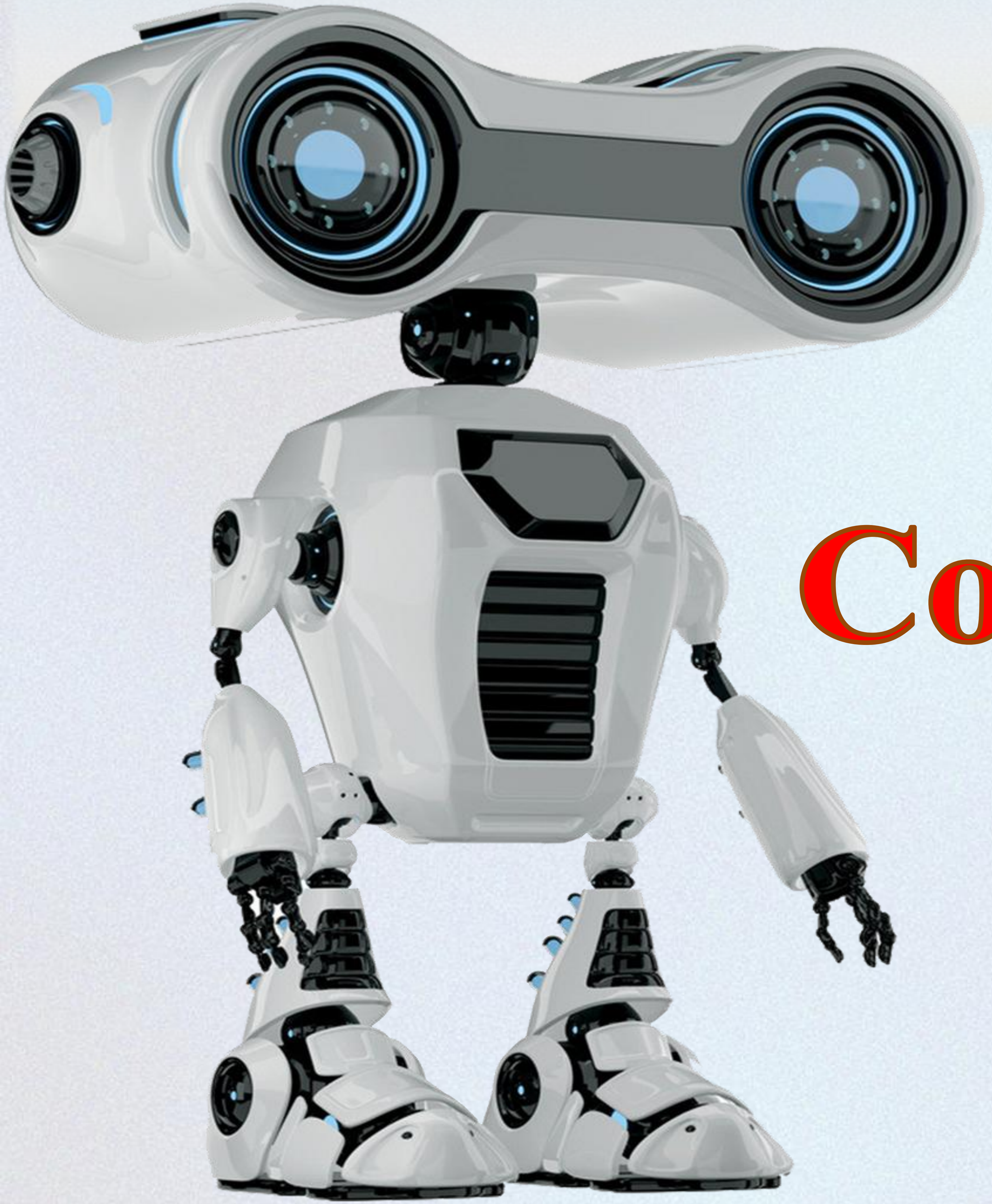
Training Process: Like other neural networks, CNNs are trained using backpropagation and gradient descent to minimize the error between predicted and actual outputs by updating the filters and weights.

Why CNNs Are Powerful: Parameter sharing (same filter used across image) and local connectivity (focus on small regions) make CNNs more efficient and effective for visual tasks.

Recurrent Neural Network

A looping constraint on the hidden layer of ANN turns to RNN. • RNN captures the sequential information present in the input data i.e. dependency be-tween the words in the text while making predictions:
• RNNs share the parameters across different time steps. This is popularly known as Parameter Sharing. This results in fewer parameters to train and decreases the computational cost





Computer Vision

Computer Vision

•What is Computer Vision?

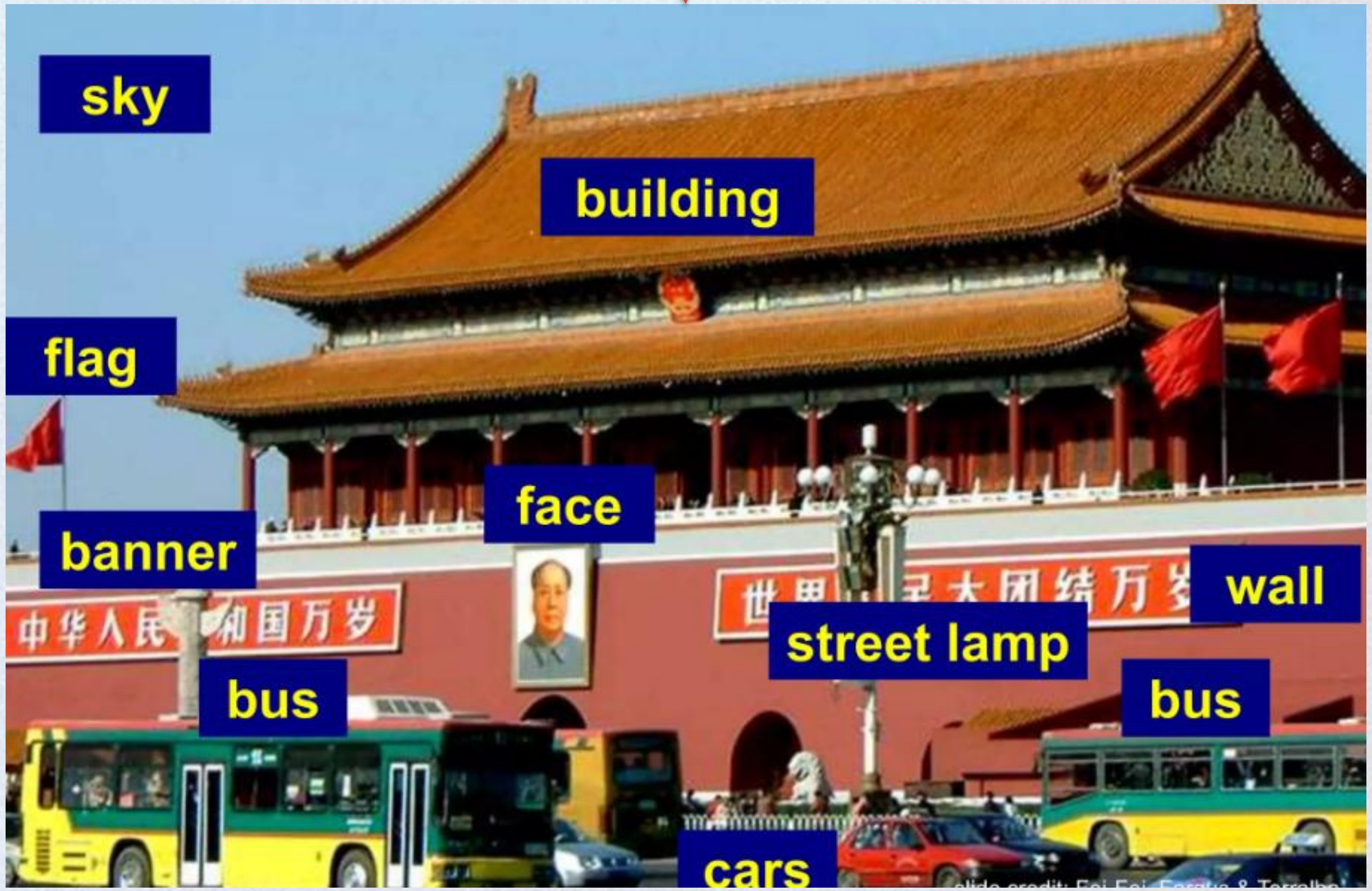
Explain it as "teaching computers to understand images and videos, just like humans do.

For example, when you look at a photo, you can tell what’s in it—a computer can learn to do the same.

•Passive vs. Active Sensing

- Passive: Cameras capture light that’s already there (like how your eyes work).
- Active: Devices like radar send out signals to "see" (like bats using sound).

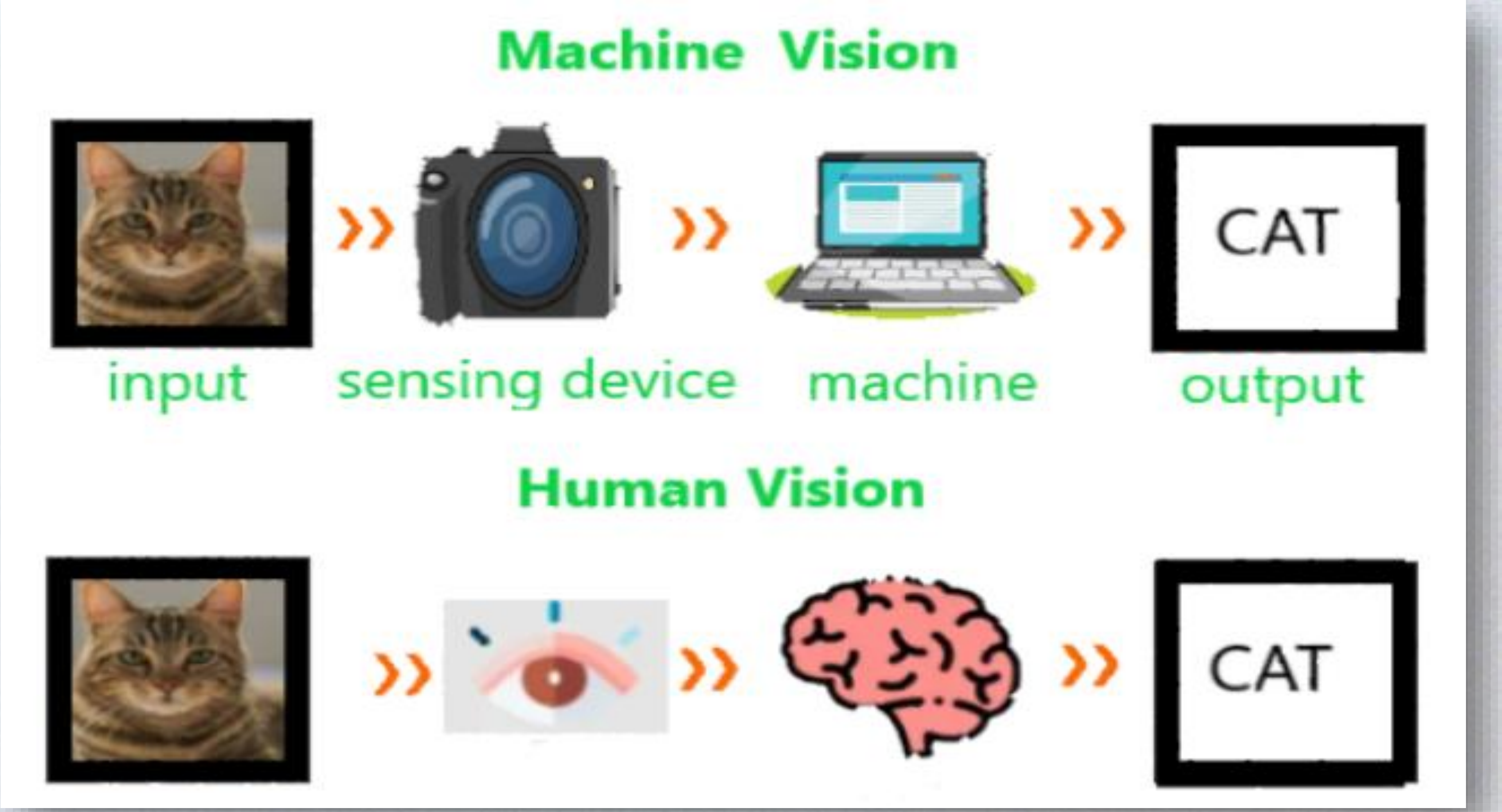
Computer Vision



Computer Vision

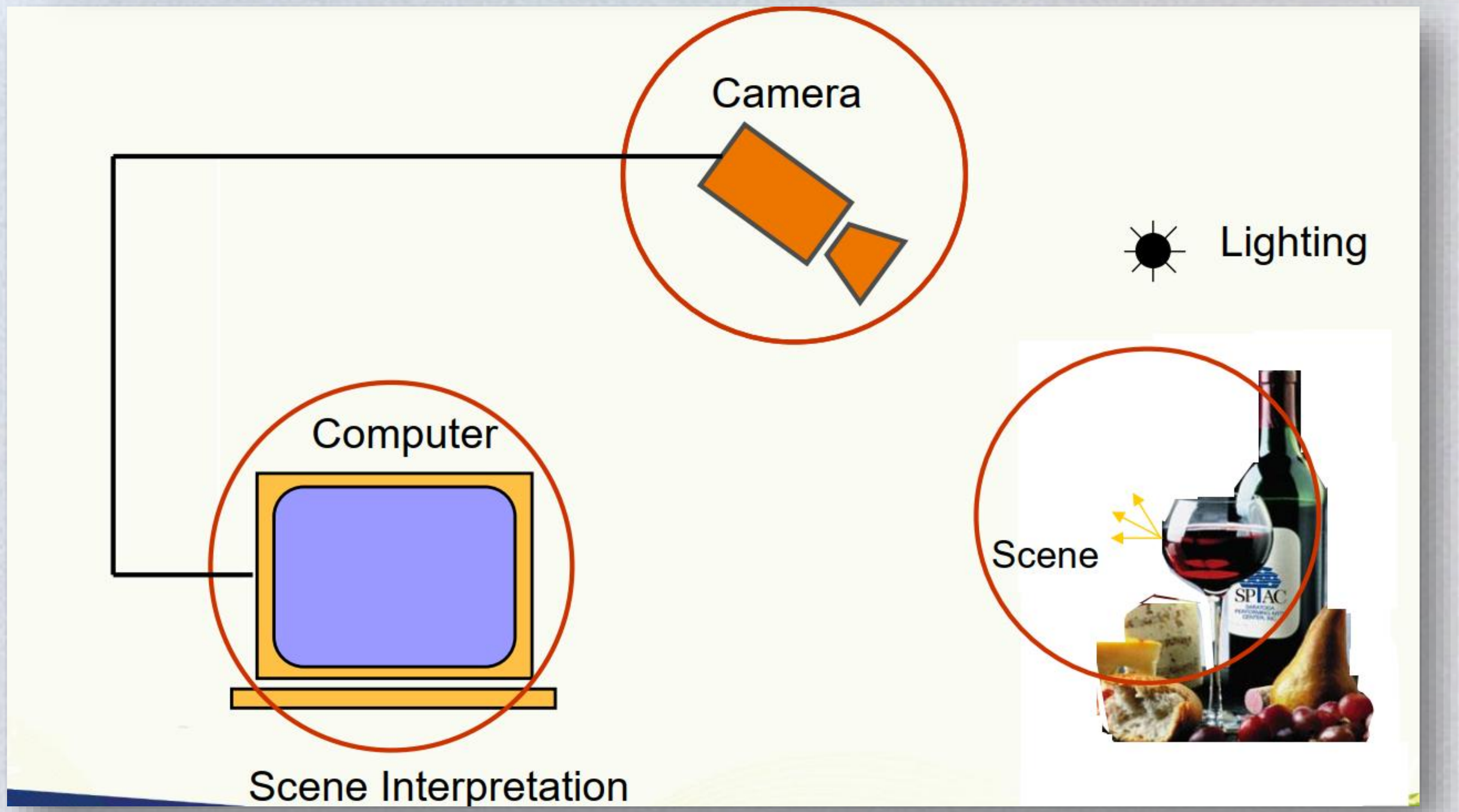
Computer Vision vs. Human Vision

Compare how humans see a cat in a photo (as a fluffy animal) vs. how a computer sees it (as a grid of numbers representing colors and shapes).



Key Components Break it down:

- Camera: The "eye" that captures images.
- Lighting: Affects how clear the image is.
- Computer: Processes the image to understand it.
- Scene Interpretation: The computer's "brain" figuring out what's in the image.



Computer Vision

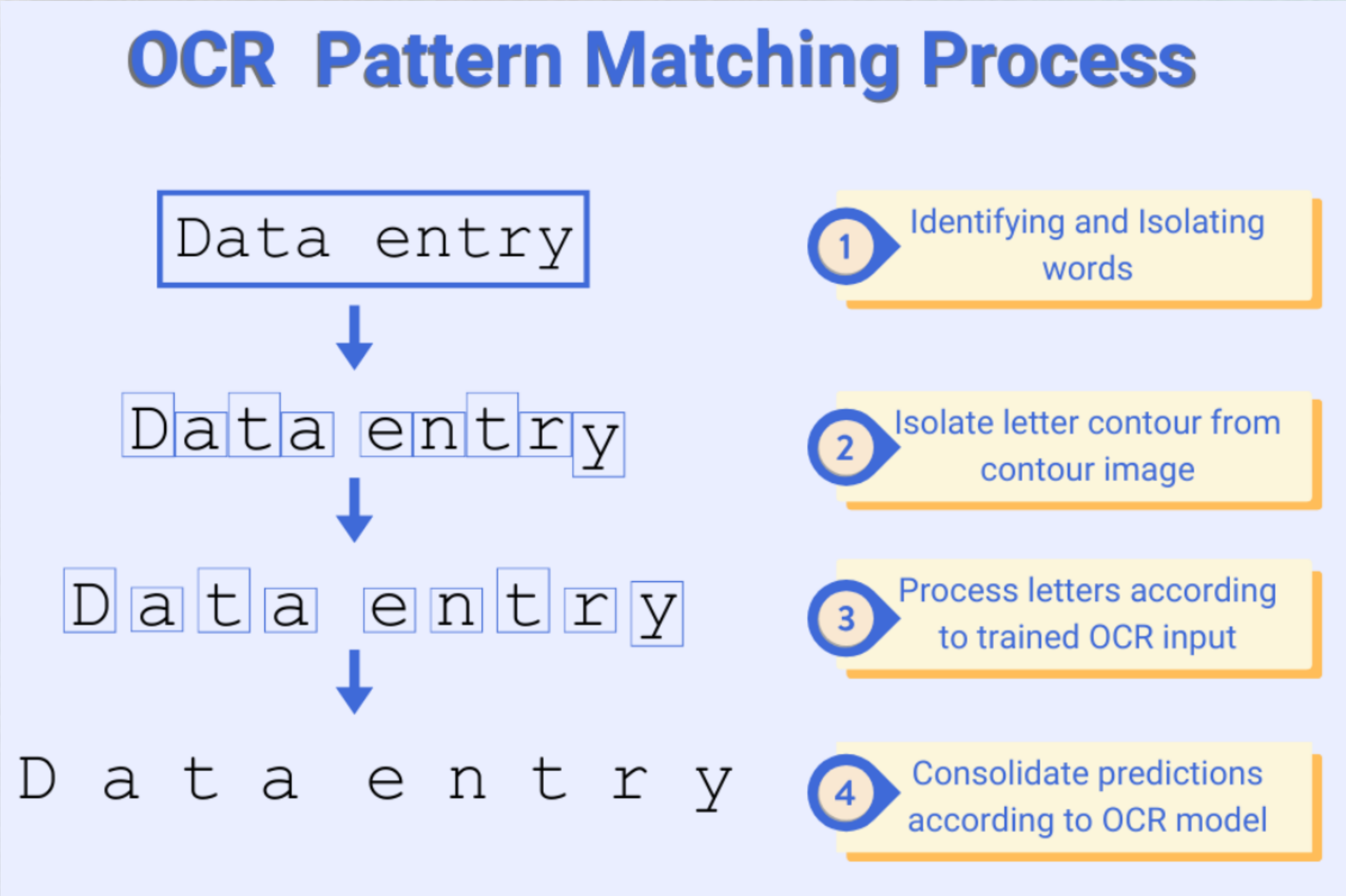
Goal of Computer Vision

- Recognize object & people
- Pattern Recognition
- Virtual Reality
- Computer Graphic
- Image enhancement (reduce noise)
- Image processing operations

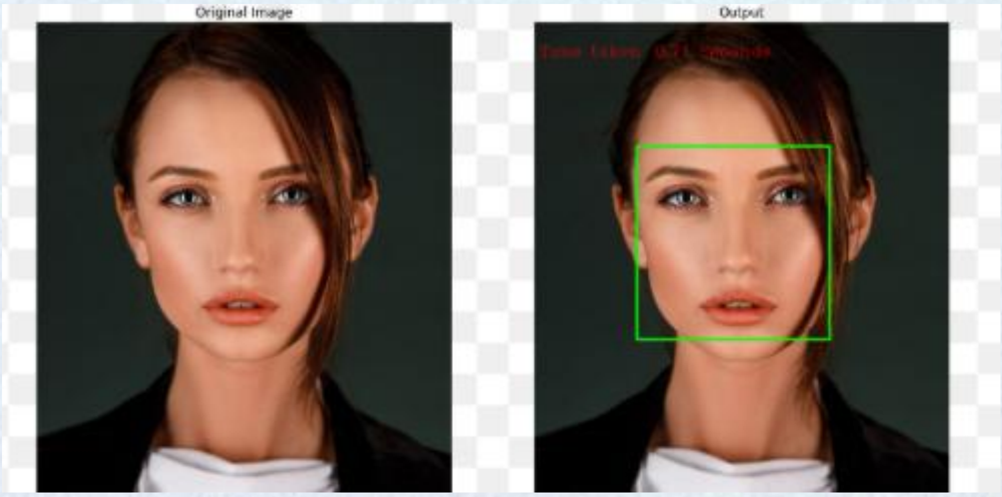
Why computer vision matters

Computer Vision (Real-World Applications)

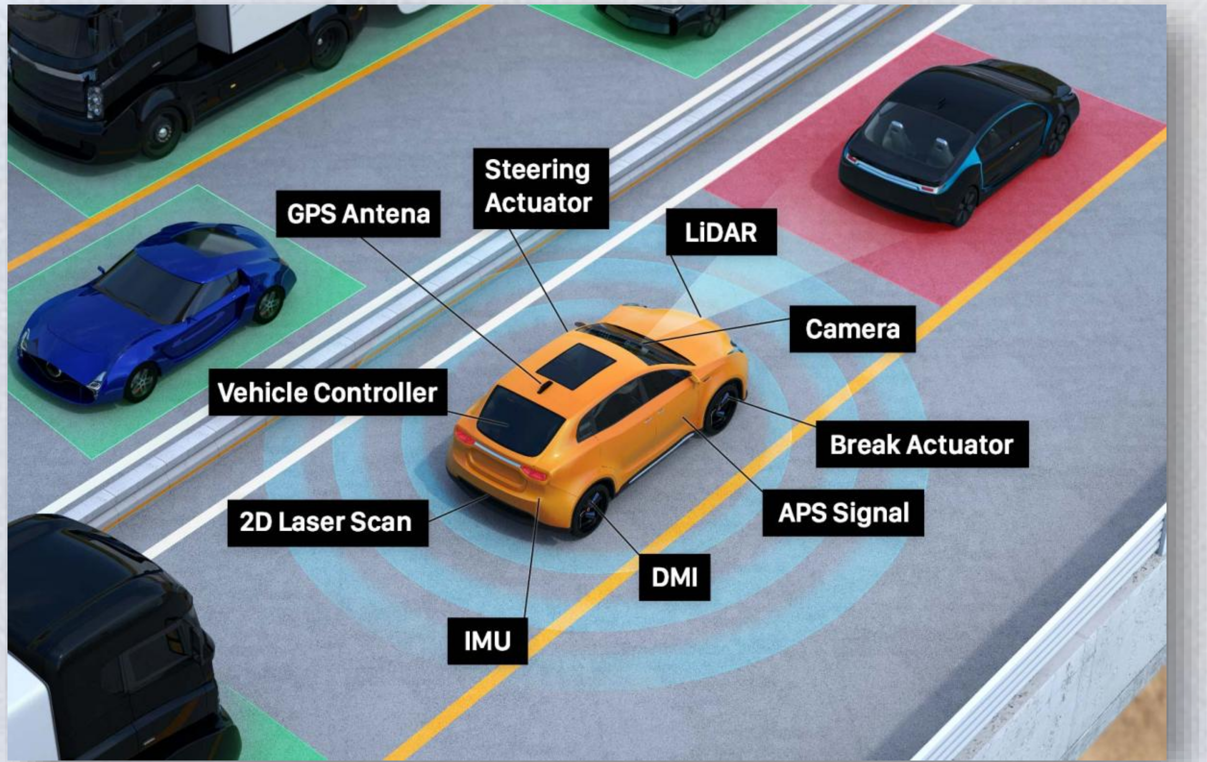
OCR : Like scanning a handwritten note and turning it into typed text.



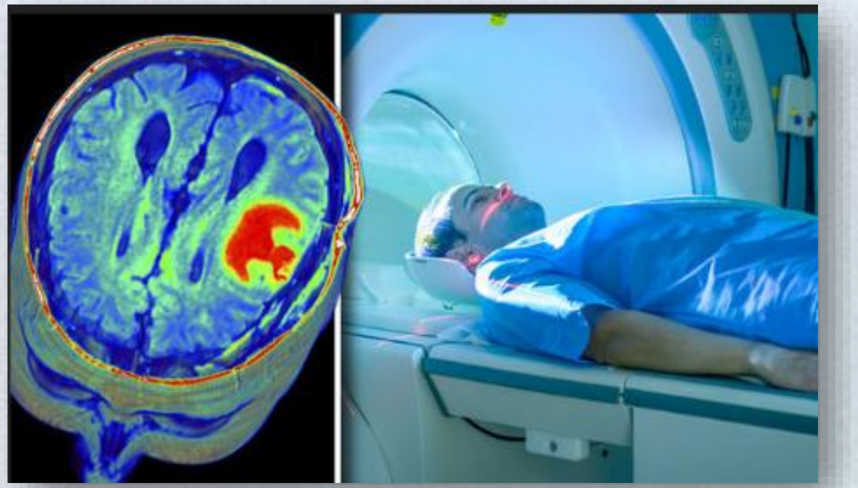
Face Detection : How your phone camera recognizes faces to focus.



Self-Driving Cars : Cars use cameras to "see" roads and avoid obstacles.



Medical Imaging : Like MRI scans helping doctors see inside the body.



Computer Vision

Core Problems Simplify the two main tasks:

- **Reconstruction**: Building a 3D model from photos (e.g., making a map from drone images).
- **Recognition**: Identifying objects (e.g., telling apart a cat from a dog in a photo).

Image plane is subdivided into a grid of a few million pixels.

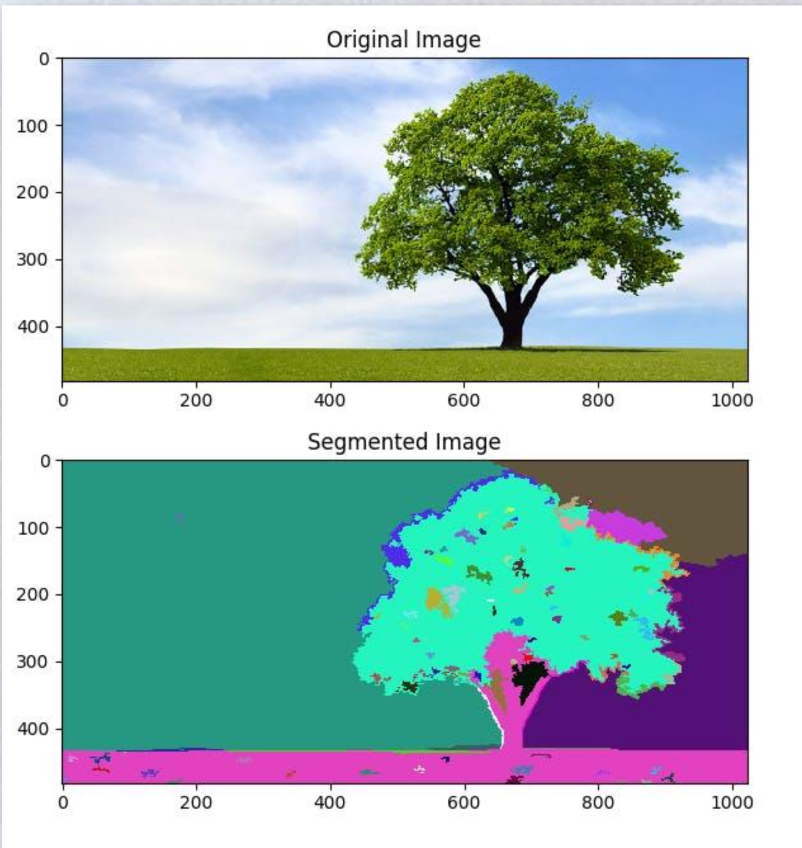
Image Features:

- A **feature** is a number we get by doing simple calculations on an image. It helps us understand important details in the image.
- **Edges** are lines or curves where the image changes sharply in brightness—like borders between objects.
- **Texture** is the visible pattern or surface detail in an image, like smooth, rough, or striped areas.

Region Segmentation:

Segmentation means dividing an image into parts that pixel look similar.

Each image pixel (region) is grouped based on things like color, brightness, or texture.



THANK YOU!