

Machine Learning: Tasks, Taxonomy & Beyond

From Classification to Deep Learning

Nipun Batra | IIT Gandhinagar

ML is Everywhere in Your Daily Life

Time	What You Do	ML Behind It
Morning	Phone unlocks with your face	Face Recognition
Commute	Google Maps predicts traffic	Time Series Prediction
Email	Gmail filters spam, suggests replies	Text Classification + Generation
Music	Spotify recommends songs	Recommendation Systems
Shopping	Amazon shows "You might also like..."	Collaborative Filtering
Photos	Google Photos groups by faces	Clustering + Image Classification
Evening	Netflix suggests what to watch	Recommendation Systems
Chat	You ask ChatGPT a question	Language Models (Generative AI)

Each of these is a different ML task!

The Big Question

Every ML task boils down to **one question**:

"**What are you trying to PREDICT?**"

Predicting a Category?

Classification

"Is this email spam?"

Predicting a Number?

Regression

"What will be the price?"

Predicting a Sequence?

Seq2Seq

"How do you say this in French?"

Creating Something New?

Generative

"Draw me a cat in space"

Once you know the "output type", you know which family the task belongs to!

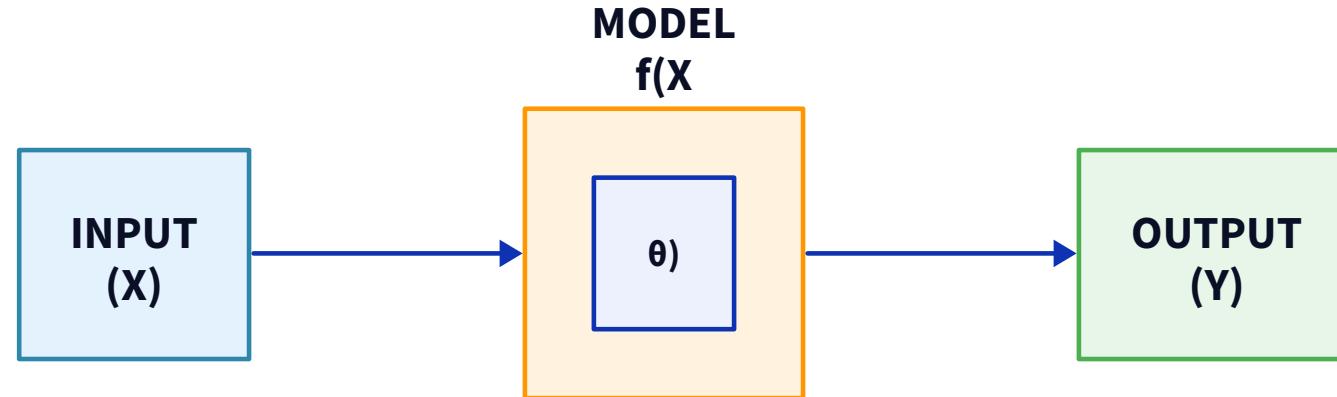
The Three Learning Paradigms

Supervised

"Learn from examples"

Data: X + Y (labels)
Tasks: Classification,
Regression, Detection

The Universal ML Recipe

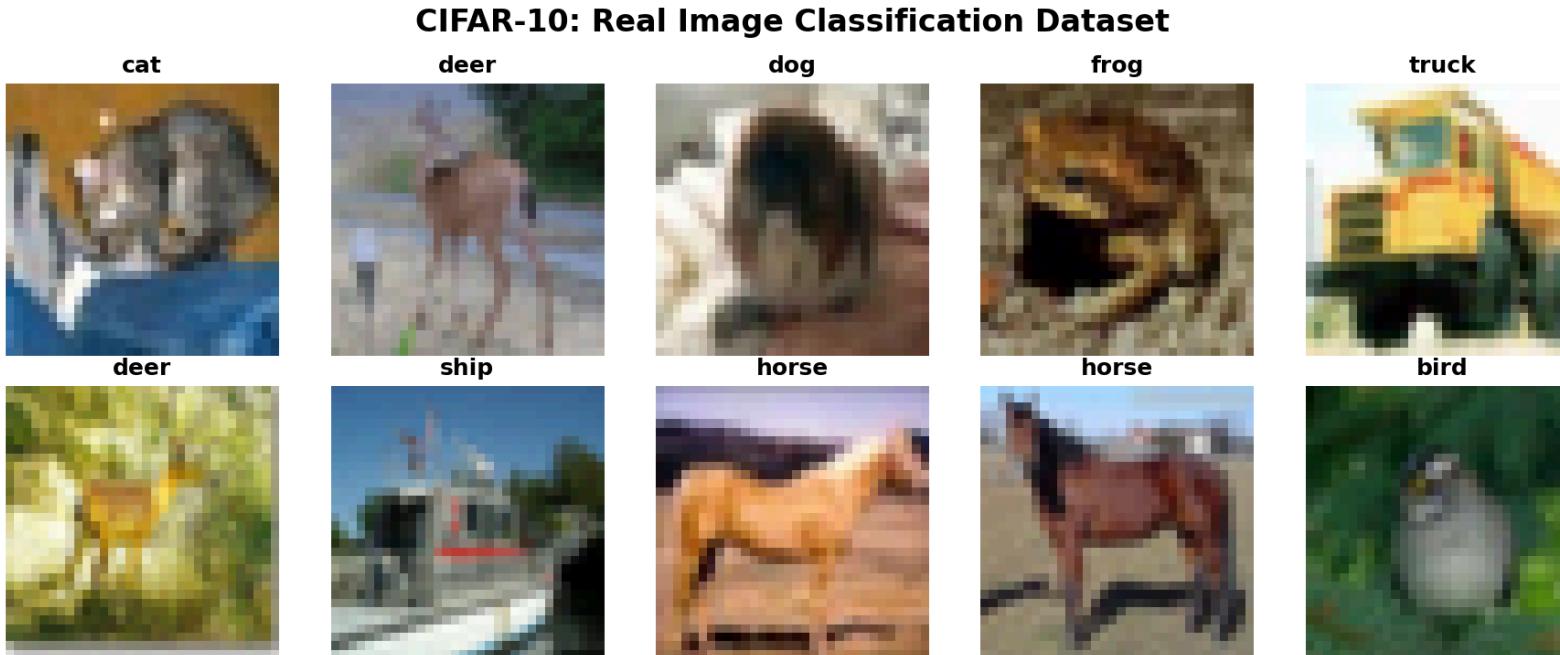


What changes between tasks:

- What **X** looks like (image, text, audio, numbers)
- What **Y** looks like (label, number, sequence, image)

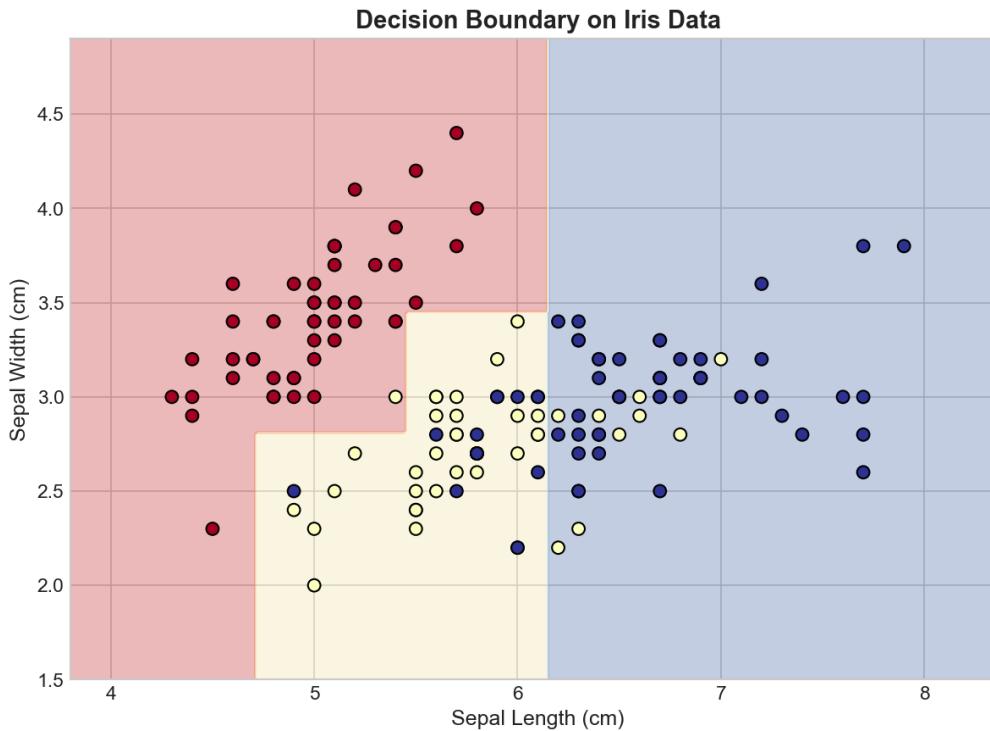
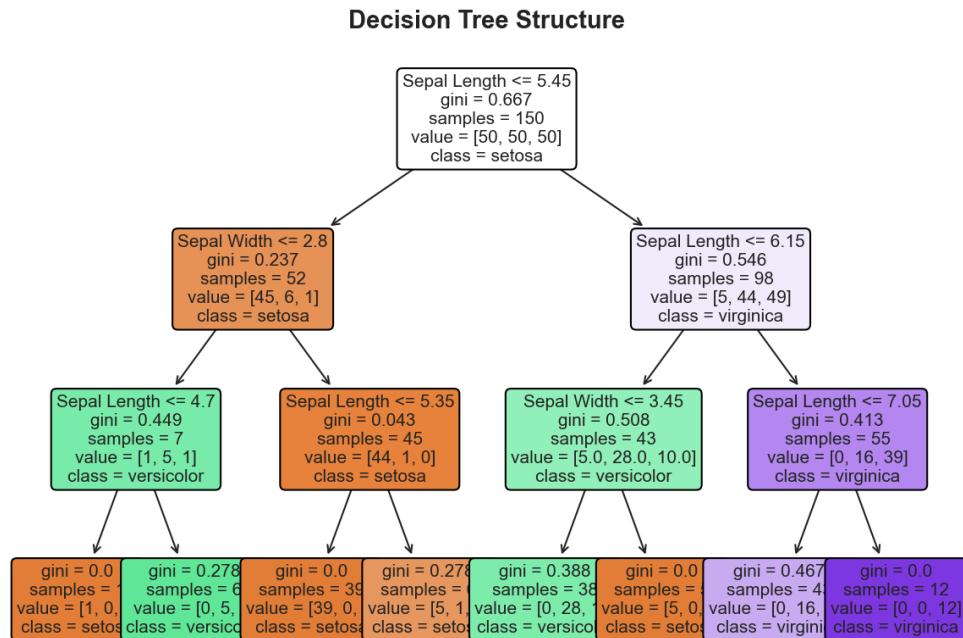
"Which Bucket Does This Belong To?"

Classification: Real Examples from CIFAR-10



You look at the input and pick **one category** from a fixed set. That's classification!

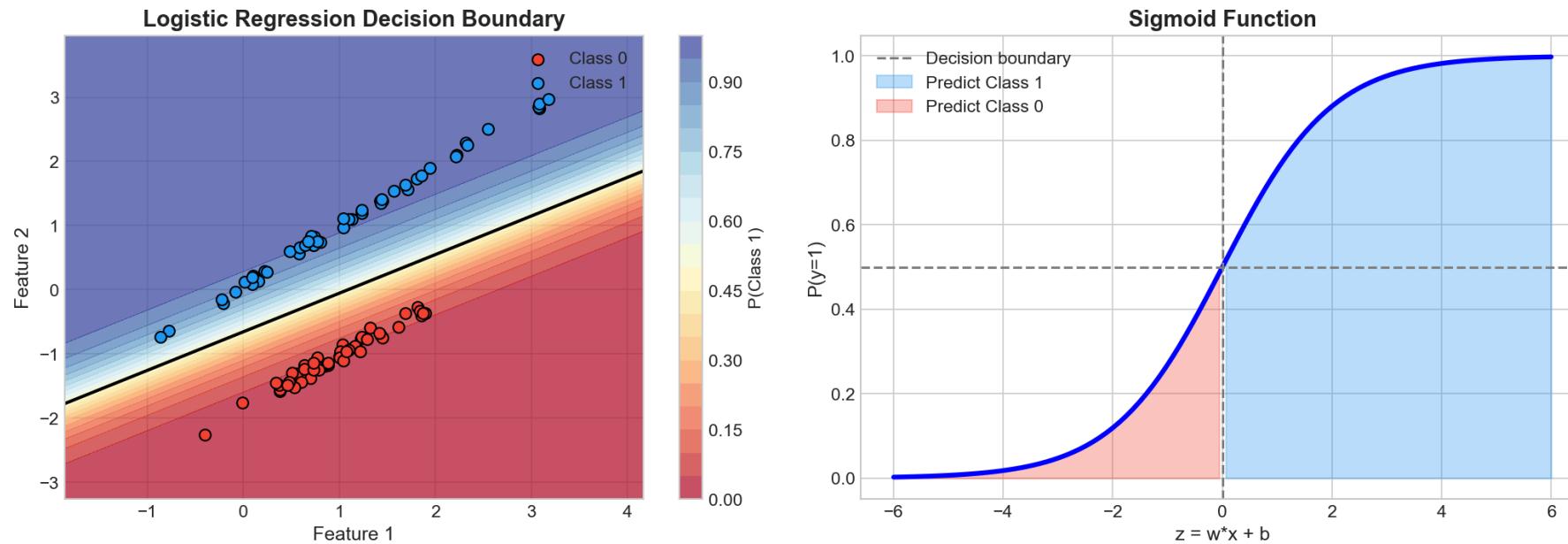
Classification: How a Decision Tree Learns



A decision tree learns **if-then rules** from data:

"If sepal length > 5.5 AND sepal width < 3.0, then iris-versicolor"

Classification: Logistic Regression



Logistic regression learns a **decision boundary** that separates classes.

Binary vs Multi-Class Classification

Binary Classification

Two possible outcomes

- Spam / Not Spam
- Fraud / Legitimate
- Pass / Fail
- Tumor: Benign / Malignant

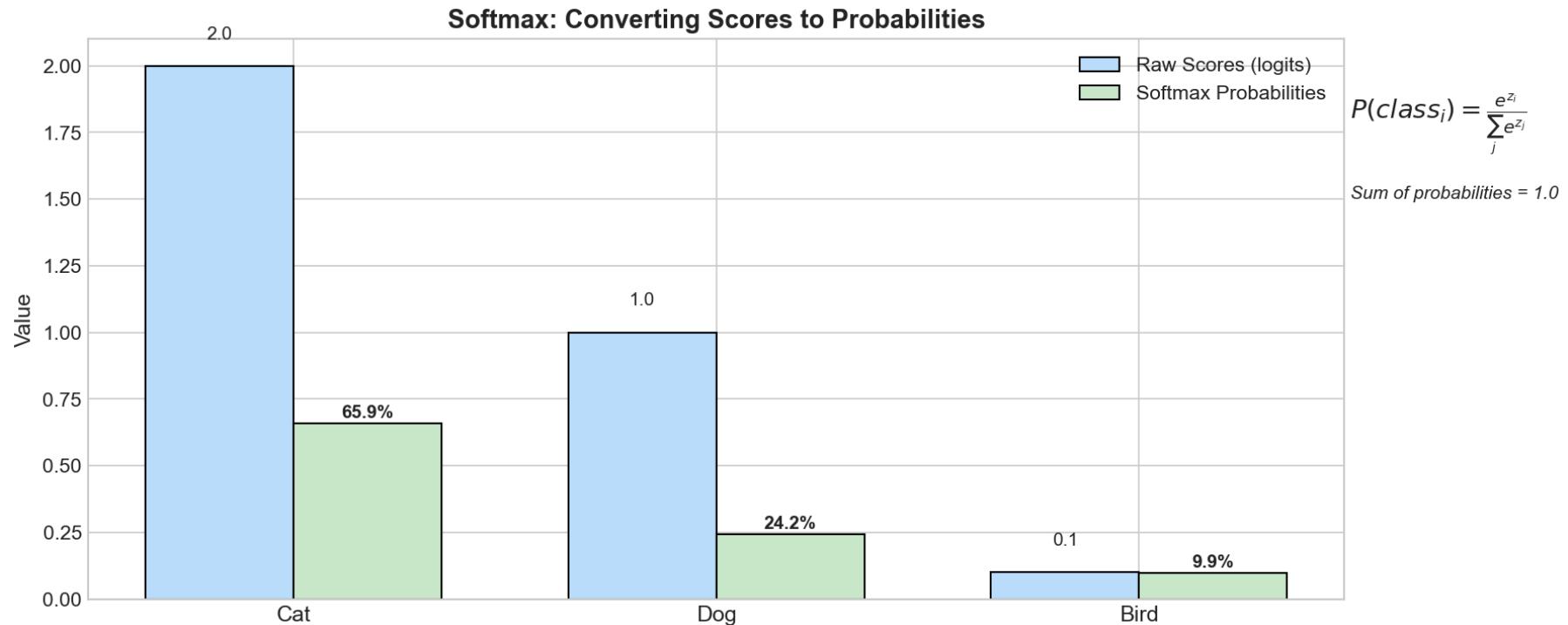
Multi-Class Classification

Many possible outcomes

- Digit recognition (0-9)
- ImageNet (1000 classes)
- Emotion detection (6+ emotions)
- Animal species identification

Same algorithm, just different number of outputs!

The Math: Softmax Turns Scores into Probabilities



Softmax converts raw scores (logits) to probabilities that sum to 1.

The model isn't just saying "Cat" - it's saying "85% sure it's a cat!"

"How Much? How Many?"

Regression: When the Answer is a Number

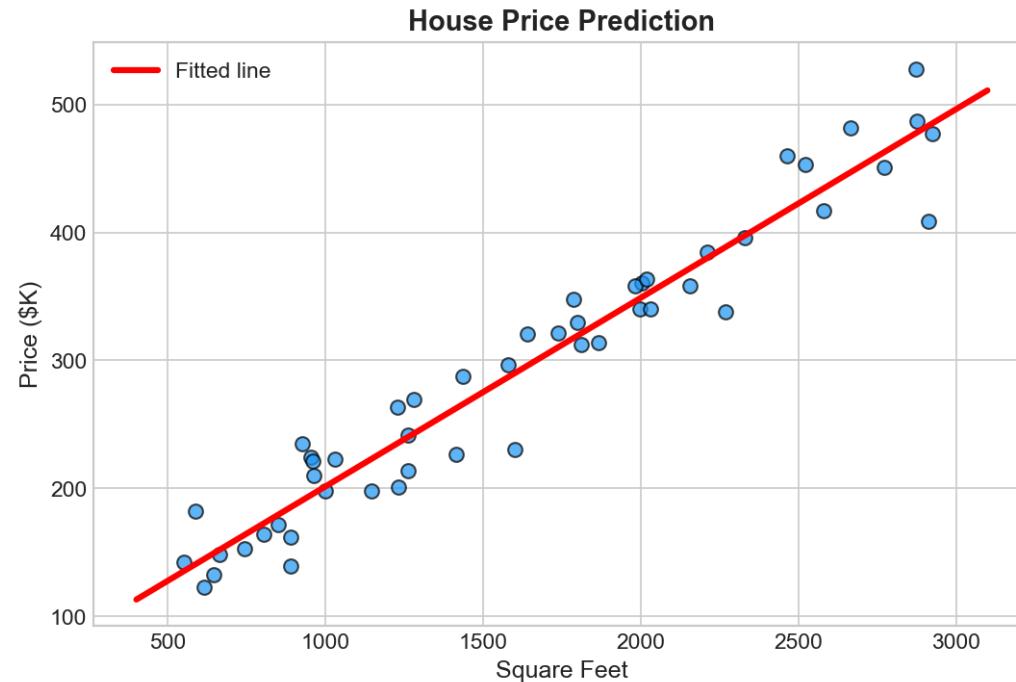
Classification: "*Which category?*" - Discrete answer

Regression: "*How much?*" - Continuous number

Question	Answer
"How old is this person?"	27.3 years
"What's this house worth?"	\$425,000
"How many units will sell?"	1,247 units
"What temperature tomorrow?"	28.5 C
"How long until the bus arrives?"	7.2 minutes

The output is **any number** on a continuous scale!

Regression in Action: Linear Regression



Linear Regression Formula:

$$\hat{y} = w_0 + w_1 \cdot x$$

Fitted: price = 54,241 + 147 * sqft

Each extra sqft adds ~\$150 to price!

The model learns: **Price = \$50,000 + \$150 * (square feet)**

Regression is Hidden Everywhere!

Bounding box detection is actually **regression**:



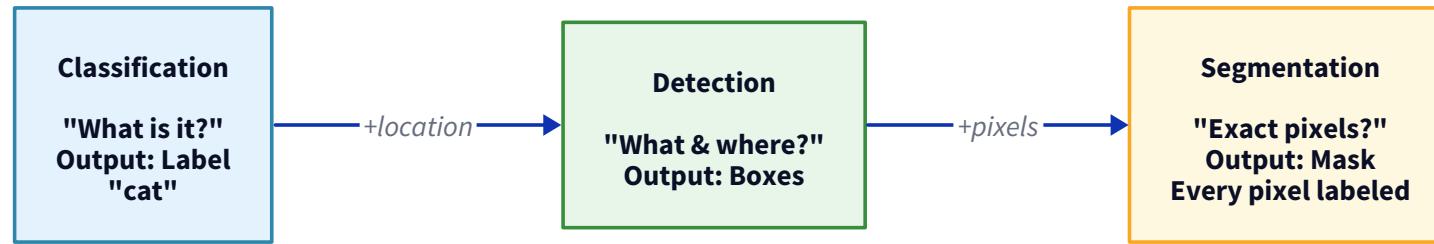
This box needs 4 numbers:

$x = 50$ (left edge) ← Regression!
 $y = 30$ (top edge) ← Regression!
 $w = 100$ (width) ← Regression!
 $h = 80$ (height) ← Regression!

DETECTION = Classification (what?) + Regression (where?)

From Labels to Pixels

The Vision Task Ladder



Each level gives you **more information** but requires **more data and compute!**

Level 1: Image Classification

What: Assign one label to an image.



Use Cases:

- Google Photos: "Show me all photos with dogs"
- Medical: "Is this X-ray normal or abnormal?"
- Quality Control: "Is this product defective?"

Level 2: Object Detection

Detection = Classification (what) + Regression (where)

Object Detection on Real Images (COCO-trained YOLOv8)

Input Image



YOLOv8 Detection



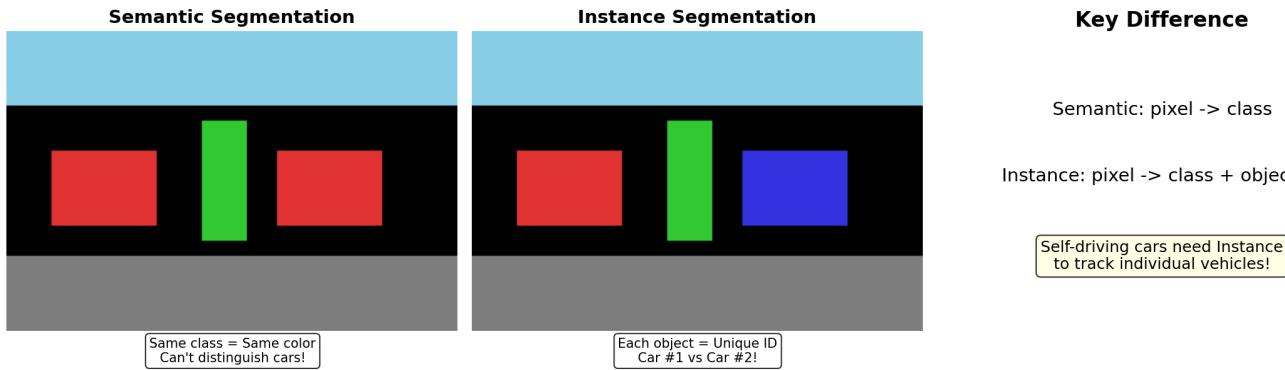
Input Image



YOLOv8 Detection



Level 3 & 4: Segmentation



Instance Segmentation in Action

Instance Segmentation: Pixel-Perfect Object Boundaries

Input Image



Instance Segmentation (YOLOv8-seg)

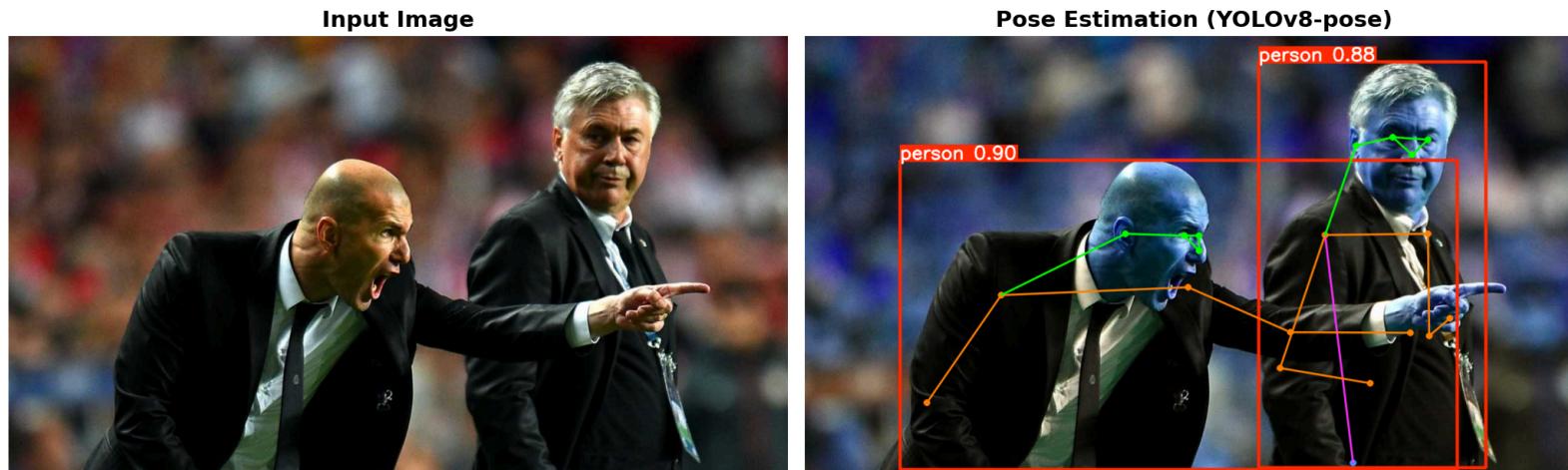


Self-driving cars need Instance Segmentation - they must track WHICH car is doing what!

Pose Estimation: Finding Body Keypoints

What: Find skeleton keypoints of humans or animals.

Human Pose Estimation: 17 Body Keypoints



Applications: Fitness apps, motion capture, sign language, fall detection

Teaching Machines to Read & Write

The NLP Task Landscape

Task Type	What It Does	Example
Sentiment Analysis	Classify emotion	"Great movie!" → Positive
Named Entity Recognition	Find names, places, dates	"Sundar Pichai visited NYC"
Question Answering	Find answers in text	"When was Einstein born?"
Translation	Convert between languages	English → Hindi
Summarization	Shorten long text	1000 words → 50 words
Text Generation	Create new text	ChatGPT, Claude

Modern LLMs (GPT-4, Claude) can do ALL of these with a single model!

Named Entity Recognition (NER)

Classify **each word** in the sequence:

Input:	"Sundar	Pichai	visited	New	York	yesterday"
	▼	▼	▼	▼	▼	▼
Output:	PER	PER	0	LOC	LOC	0

PER = Person Name

LOC = Location

0 = Other (not an entity)

Think of it as "semantic segmentation for text" - every word gets a label!

Finding Patterns Without Labels

Supervised vs Unsupervised

SUPERVISED:

| Data: X (features)
| Labels: Y (answers)

| Learn: $f(X) \rightarrow Y$

| "Teach by example"

UNSUPERVISED:

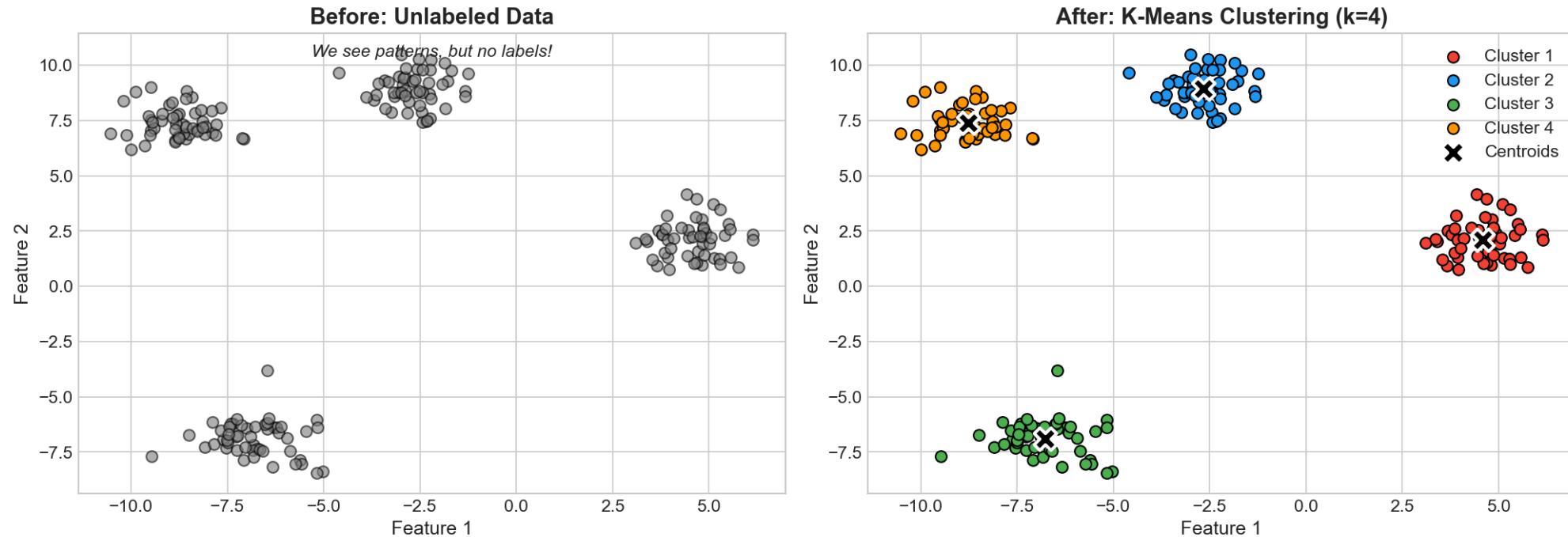
| Data: X (features)
| Labels: NONE!

| Find: patterns in X

| "Learn by exploration"

No one tells the model what to look for - it discovers structure on its own!

Clustering: K-Means in Action



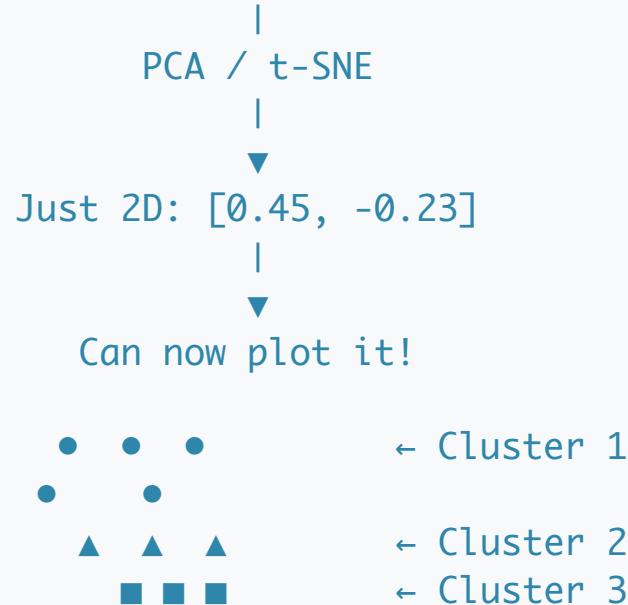
K-Means: No labels needed! The algorithm discovers natural groupings.

Applications: Customer segmentation, gene expression analysis, document clustering

Dimensionality Reduction

Problem: High-dimensional data is hard to visualize.

Original: 1000-dimensional data
(Can't visualize 1000 axes!)



Creating New Data

Generative vs Discriminative

DISCRIMINATIVE (Classification):

[Image of cat] → Model → "Cat" or "Dog"

Given X, predict Y. "What IS this?"

GENERATIVE (Creation):

"Draw a cat" → Model → [NEW image of a cat!]

Create NEW X from scratch. "Make something like this"

The Generative AI Revolution

Domain	Tool	What It Does
Text	ChatGPT, Claude	Write essays, code, poems
Images	DALL-E, Midjourney, Stable Diffusion	Generate any image from text
Music	Suno, Udio	Create full songs with lyrics
Video	Sora, Runway	Generate realistic video clips
Code	GitHub Copilot, Claude	Write and debug code
Voice	ElevenLabs	Clone and synthesize voices

All of these generate NEW content that never existed before!

Combining Everything

Multimodal = Multiple Modalities

Modalities: Text, Image, Audio, Video, etc.

SINGLE-MODAL:

Image → Model → Cat
(just images)

Text → Model → Sent
(just text)

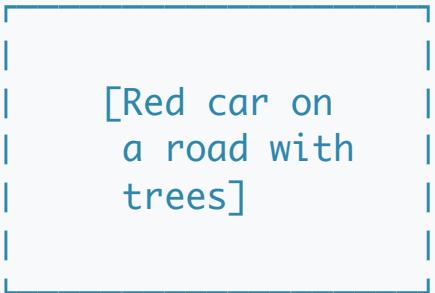
MULTI-MODAL:

Image + Question → Model → Answer
[Photo of 3 dogs]
"How many dogs?"
↓
"Three"

Modern AI (GPT-4, Claude, Gemini) is multimodal - it can see AND read AND hear!

Visual Question Answering (VQA)

Image:



Questions & Answers:

Q: "What color is the car?"

A: "Red"

Q: "Is it daytime or night?"

A: "Daytime"

Q: "How many trees are visible?"

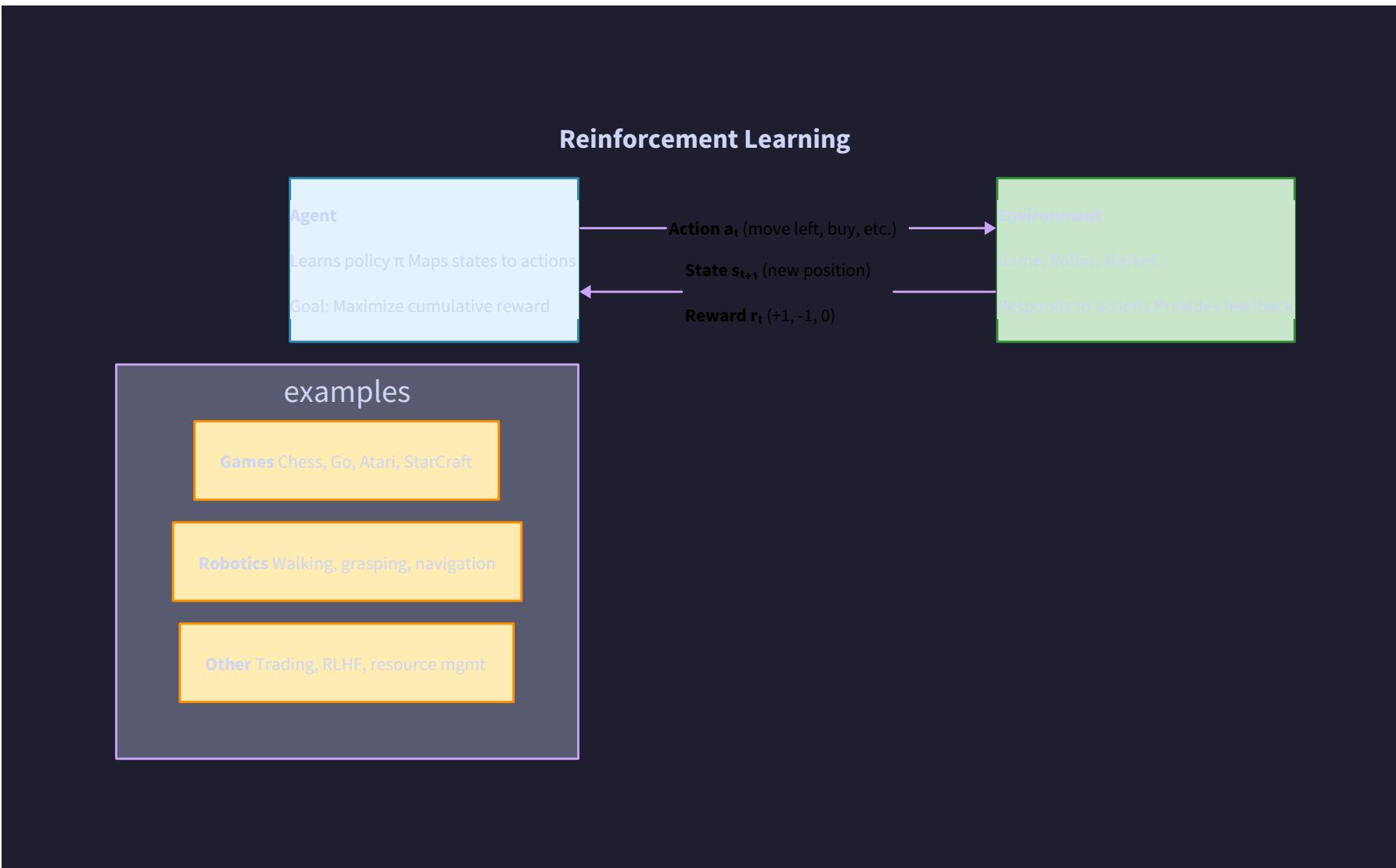
A: "Four trees"

Requires BOTH:

- Understanding image
- Understanding language
- Reasoning about both!

Learning Through Interaction

RL: A Different Paradigm



Goal: Maximize total reward over time through trial and error

RL Examples

Domain	Example	What It Learned
Games	AlphaGo	Beat world champion at Go
Games	AlphaStar	Grandmaster at StarCraft II
Robotics	Boston Dynamics	Walk, run, dance
Infrastructure	Google Data Centers	40% energy reduction
AI Alignment	RLHF for ChatGPT	Be helpful and safe

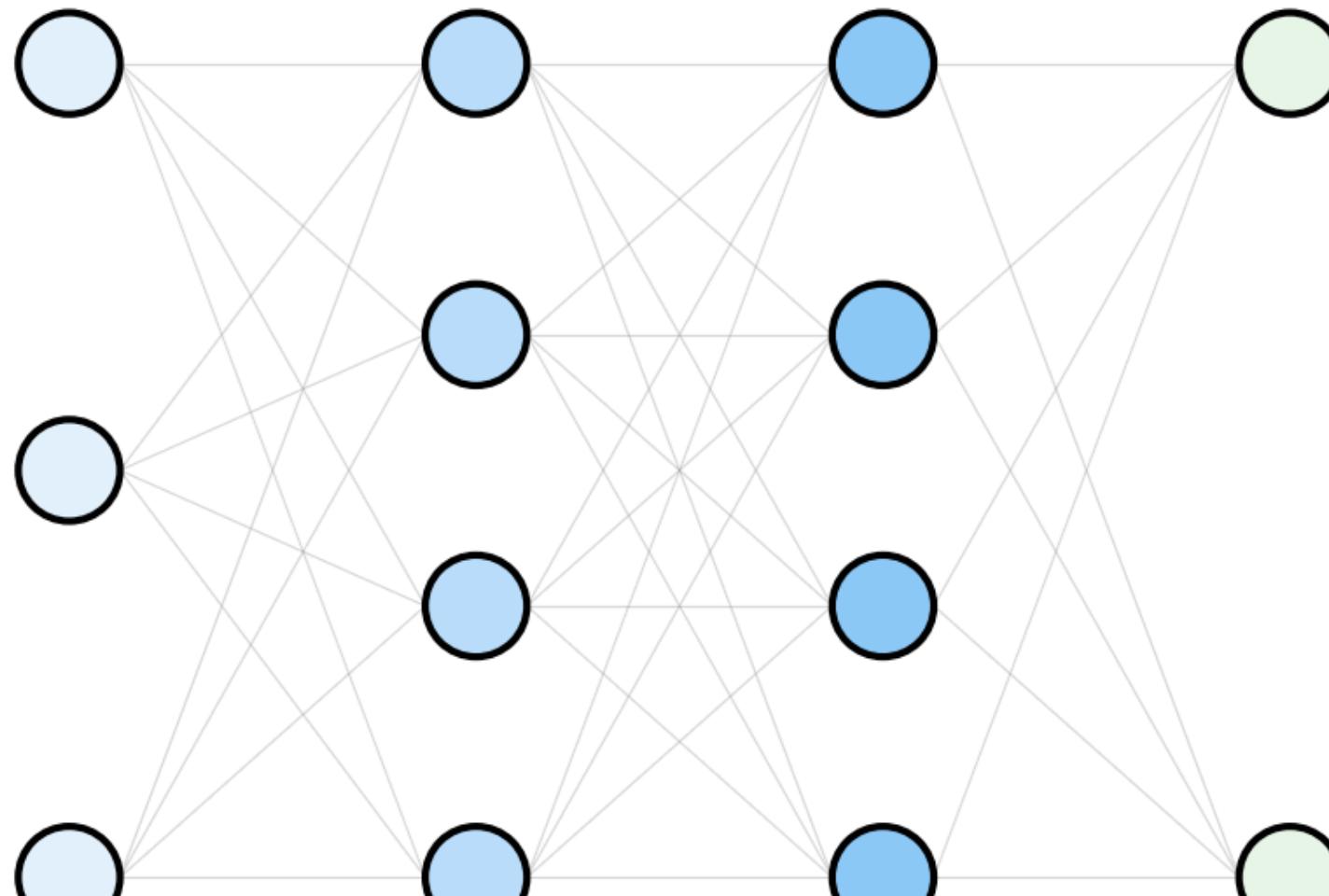
RLHF (Reinforcement Learning from Human Feedback) is how ChatGPT learns to be helpful!

Neural Networks & Deep Learning

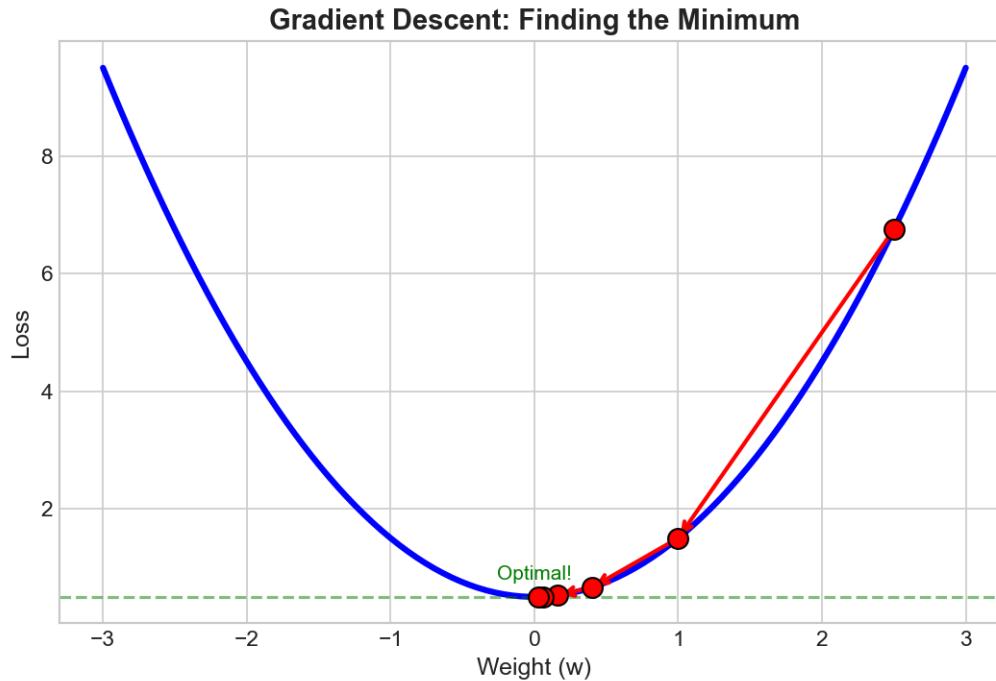
Neural Networks: The Universal Tool

Neural Network: Universal Function Approximator

Each connection has a learnable weight



How Neural Networks Learn: Gradient Descent



Gradient Descent Update Rule:

$$w_{\text{new}} = w_{\text{old}} - \eta \cdot \nabla L$$

η = learning rate (step size)

∇L = gradient (direction of steepest ascent)

We go **OPPOSITE** to gradient to minimize loss!

Training = Finding the weights that minimize the loss function

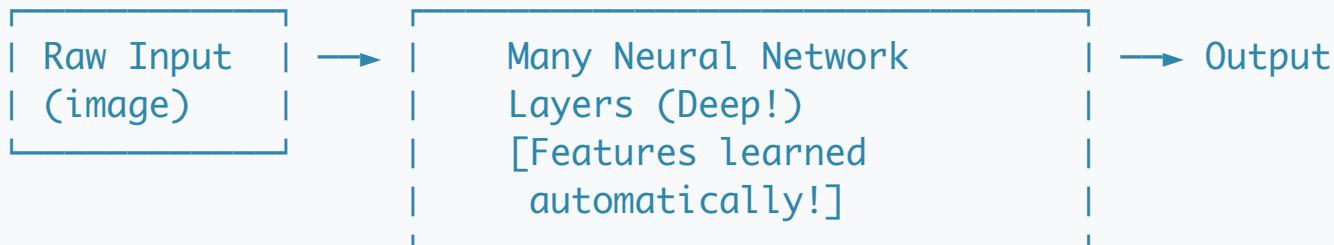
The Deep Learning Revolution

WHY "DEEP" LEARNING?

Traditional ML:

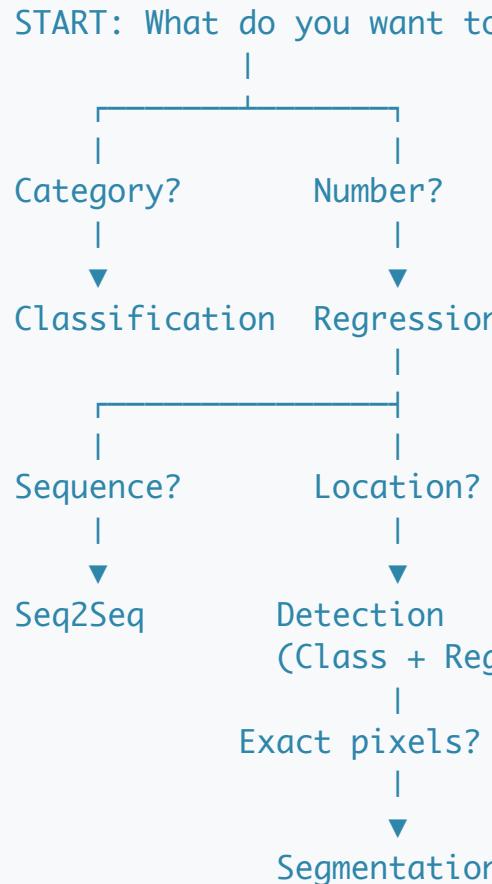


Deep Learning:



Deep learning learns the features automatically - no hand-engineering needed!

The Decision Flowchart



No labels available? → Unsupervised (Clustering, etc.)

Want to create new data? → Generative

Learning from trial/error? → Reinforcement Learning

Key Takeaways

1. **Classification** - Predict a category (discrete)
2. **Regression** - Predict a number (continuous)
3. **Detection** - Classification + Box Regression
4. **Segmentation** - Classification for every pixel
5. **Seq2Seq** - Sequence in, sequence out
6. **Unsupervised** - Find patterns without labels
7. **Generative** - Create new data
8. **Multimodal** - Combine text, images, audio
9. **RL** - Learn from rewards through interaction

Understanding the output type tells you which family of techniques to use!

Coming Up Next

Lecture 3: Language Models

- Next Token Prediction
- Pre-training, SFT, RLHF
- From GPT to ChatGPT

Lecture 4: Object Detection

- YOLO and beyond
- Real-time detection

Thank You!

"All models are wrong, but some are useful." - George Box

The key is matching the right model to the right task!

Questions?
