Deep Learning

From Fundamentals to Research-Driven

Equipping You with Research Depth and Industry Skills

By:

Dr. Zohair Ahmed

www.youtube.com/@ZohairAI Subscribe

www.begindiscovery.com

Development Environment Setup

1. Install Python (Choose One Option)

- **Option 1:** Download & install **Python** from python.org
- **Option 2:** Install **MiniConda** or **Anaconda (Full)**
- **Recommendation:** Use **Manual Python**

2. Create a Virtual Environment

- Always create a **venv** for each project
- Keeps dependencies clean and avoids version conflicts
- Activate **venv** → install packages inside it

3. Use PIP for Package Management

- pip install package_name
- Keep packages updated
- Manage requirements through requirements.txt

4. Start with Google Colab

- Best for beginners
- Free GPU
- No installation required
- Easier than Jupyter Notebook

5. Use Proper IDE for Real Projects

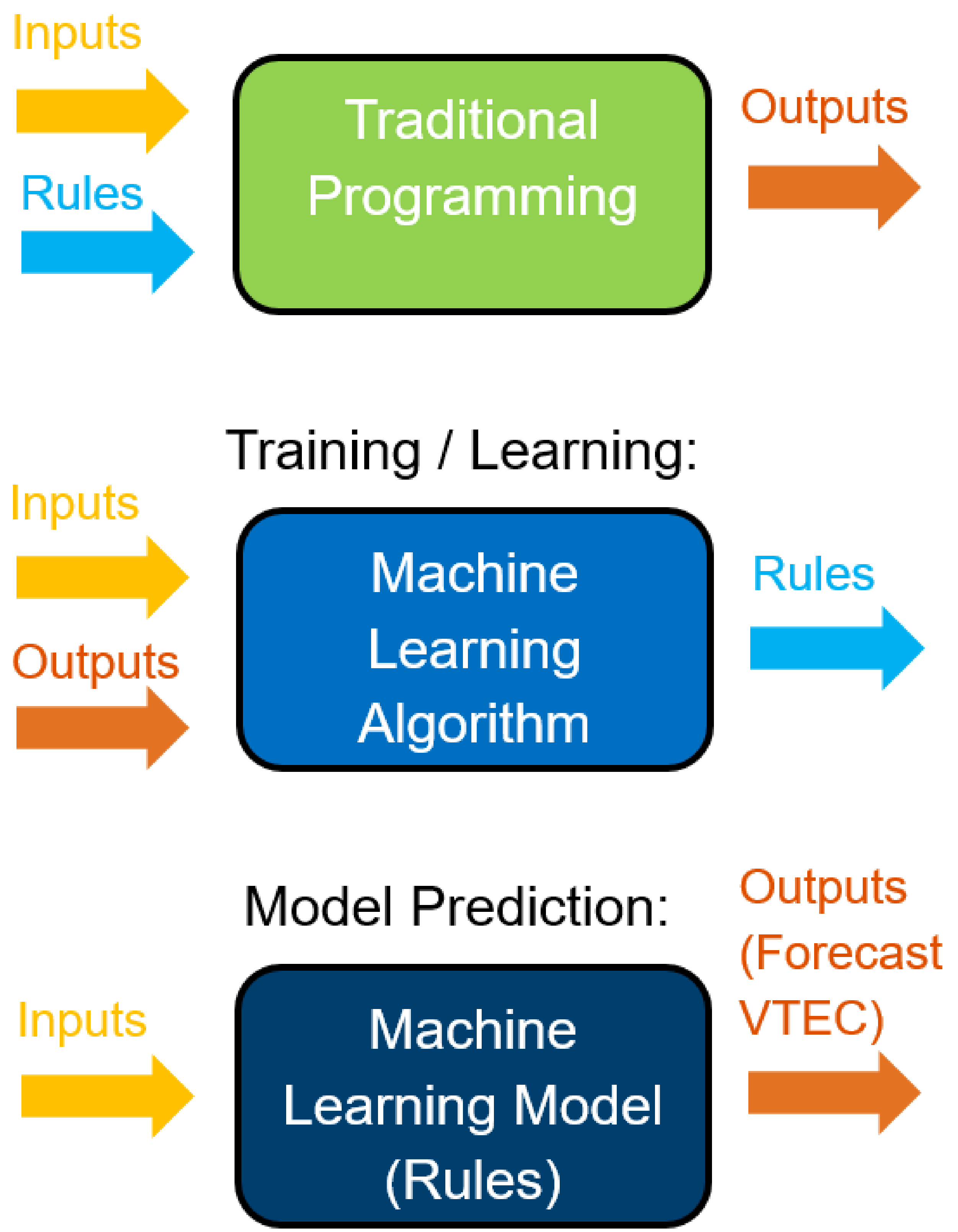
- **VS Code** (recommended)
- or **Spyder**

6. Build a Kaggle Portfolio

- Join competitions
- Publish notebooks
- Improve skills + visibility
- Essential for **Data Scientist career**

Why Study Learning Systems?

- **Computing has evolved from:**
  - *Explicit programming* → hand-crafted rules
  - To *learning from data* → models discover patterns
- **Why learning is essential:**
  - Many tasks are too complex to hand-code (e.g., vision, language)
  - Data is abundant; rules are not
  - Adaptive systems can improve over time
- **Goal:** Build systems that **improve performance** with **experience**.



Traditional Programming Example

- **Task:** Classify whether a number is even or odd.
- We write rules ourselves:
- **Input:** number
- **Rules:** written by programmer (n % 2 == 0)
- **Output:** “Even” or “Odd”

```
def is_even(n):
    if n % 2 == 0:
        return "Even"
    else:
        return "Odd"
```

Machine Learning Example

- **Task:** Predict house prices
- *Here, we do NOT write rules.*
- ML learns rules from data.
- **We provide:**
- **Input:** house size, rooms, location
- **Output:** actual price
- ML finds patterns = “rules”
- The ML algorithm produces something like:
- $\text{Price} = w \times \text{Size} + b \rightarrow \quad 0.015 \times \text{Size} - 3$

Size (sqft)	Price
1000	12 lac
1500	18 lac
1800	25 lac

Everyday Examples of Learning Systems

- **In your daily life:**
- Recommendation systems: YouTube, Netflix, Spotify
- Smart assistants: Siri, Alexa, Google Assistant
- Social media feeds: ranking, content moderation
- Navigation apps: traffic prediction, route planning
- **Key message:** Learning-based methods already mediate much of your digital experience.

What is Artificial Intelligence (AI)?

- **Informal definition:**
  - *AI = Systems that perform tasks that, if done by humans, would require intelligence.*
- **Typical AI capabilities:**
  - Perception (seeing, hearing)
  - Reasoning and planning
  - Learning and adaptation
  - Natural language understanding and generation
- **Subfields:**
  - Knowledge representation, search, planning, robotics, ML, NLP, etc.

Historical Approaches to AI

- **Symbolic / Rule-based AI:**
- Expert systems, logic rules, search-based planning
- Strength: explicit reasoning, interpretability
- Weakness: brittle, hard to scale to messy real-world data
- **Statistical AI:**
- Probability, graphical models, pattern recognition
- Strength: handles uncertainty, data-driven
- Setup for **machine learning** and later **deep learning**.

What is Machine Learning (ML)?

- **Core idea:**
- ML is about **learning a function** from data, rather than hand-coding it.
- **Classic definition (Mitchell):** *A computer program is said to learn from experience $\boldsymbol{E}$ with respect to some class of tasks $\boldsymbol{T}$ and performance measure $\boldsymbol{P}$, if its performance at tasks in $\boldsymbol{T}$, as measured by $\boldsymbol{P}$, improves with experience $\boldsymbol{E}$.*
- **Now, the Super Simple Version**
- Imagine you have a friend named **Robo**.
- **Task (T):** What you want Robo to do → e.g., recognize cats in pictures.
- **Experience (E):** The practice Robo gets → e.g., showing Robo many pictures of cats and not-cats.
- **Performance Measure (P):** How you judge Robo's progress → e.g., how many pictures Robo gets right.
- **What learning means:**
- **Robo is learning if:**
  - After seeing more examples (experience **E**),
  - Robo gets better at the task (task **T**),
  - And the score you measure (performance **P**) improves.
- **Even simpler:**
- **Learning =** getting better at something by practicing.
- A program is **"learning"** if it practices something and its results improve.

Example: Machine Learning (ML)

- **Example:**
- **Task (T):** Predicting tomorrow’s temperature
- **Experience (E):** Past weather data
- **Performance (P):** How close its predictions are to the real temperature
- If the computer predicts better and better as it sees more data → **It has learned!**

- **Key components:**
  - Data (examples)
  - Model (hypothesis space)
  - Objective / loss function
  - Learning algorithm (optimization)

What is Deep Learning (DL)?

- Deep Learning is a part of Machine Learning where we use **deep neural networks**, meaning **neural networks with many layers**.
- Think of it like teaching a child to recognize things:
  - First they learn **lines**
  - Then **shapes**
  - Then **eyes, nose, ears**
  - Then **faces**
  - Then **who the person is**
- Neural networks do something similar.
- **Why do we call it *deep*?**
- Because the network has **many layers**, one after another like a tall sandwich.
- **These layers learn hierarchically:**
- **Low-level layers** → simple things
  - edges
  - lines
  - corners
- **Mid-level layers** → patterns
  - textures
  - shapes
- **High-level layers** → big concepts
  - faces
  - cars
  - words
  - meaning in a sentence
- This "stacking" makes it powerful.

What is Deep Learning (DL)?

- **Machine Learning (ML):** A broad field: models learn patterns from data to make predictions/decisions.
  - Includes: linear/logistic regression, decision trees, random forests, gradient boosting, SVMs, kNN, Naive Bayes, clustering, etc.
- **Neural Networks (NN):** One family of ML models made of layers (linear transform + non-linear activation).
- **Deep Learning (DL):** Neural networks with **many layers** (deep architectures), plus training tricks and large-scale data/compute.

Why Deep Learning is so good?

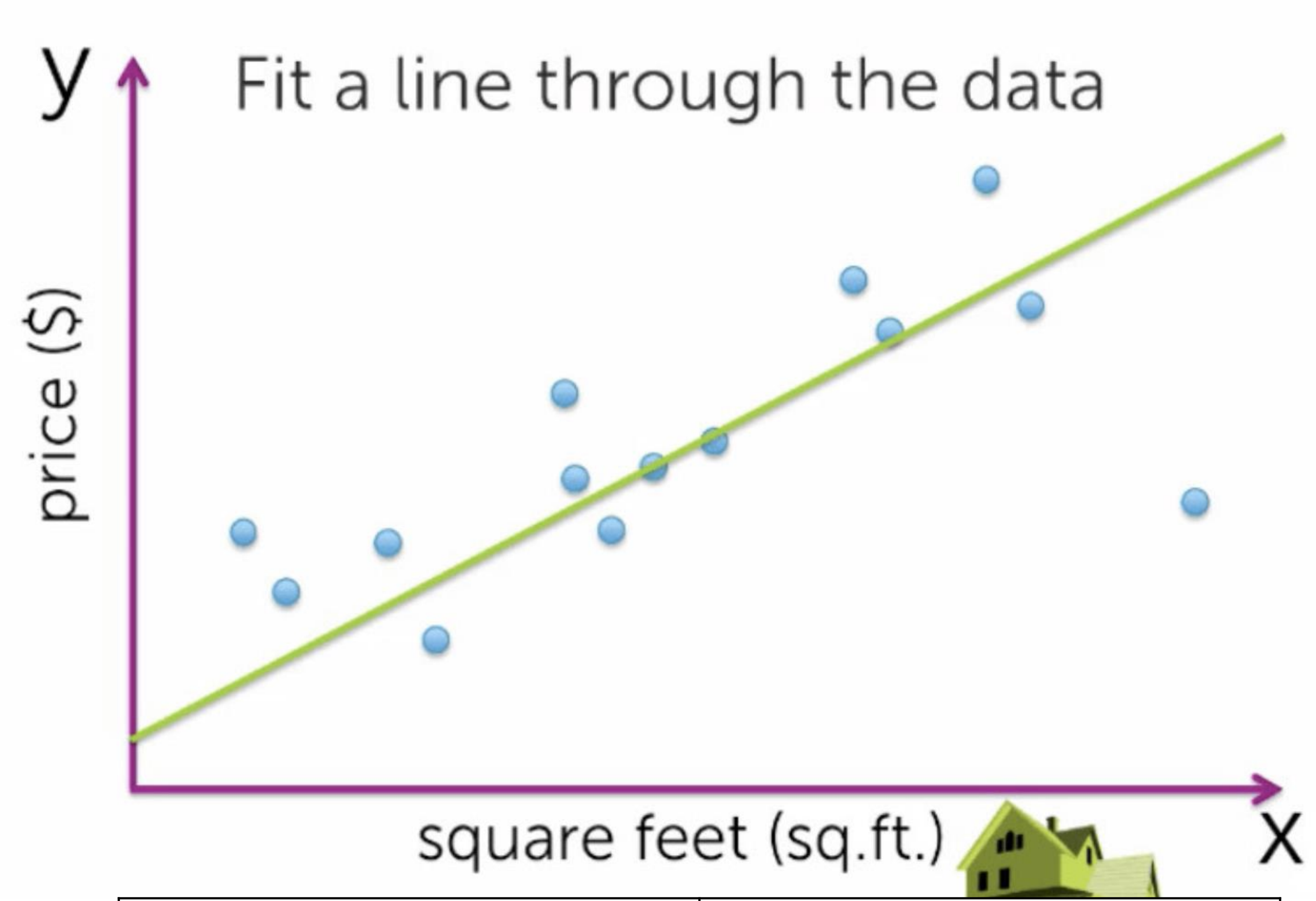
- **It learns directly from raw data**
- No need for manual feature engineering like in classic ML.
- Examples:
  - Raw pixels → image classifier
  - Raw audio → speech recognizer
  - Raw text → large language model

- **It gives state-of-the-art performance**
- Computer vision
- NLP
- Speech recognition
- Recommendation systems
- Autonomous vehicles
- Medical imaging
- Because multiple layers allow **complex patterns** to be captured easily.

Linearity

Linear Data

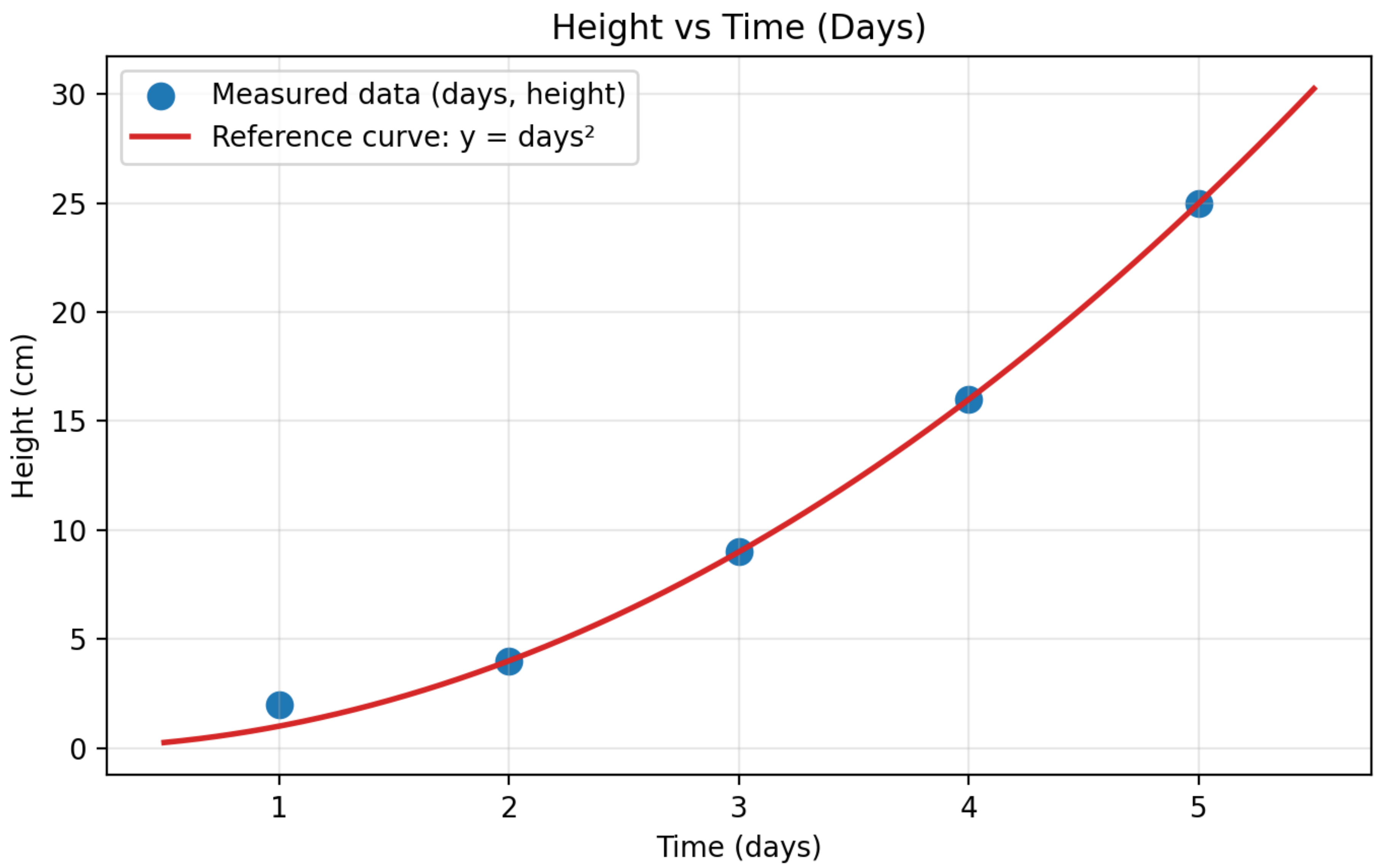
- When we say data is **linear** or **non-linear**, we're talking about **how the features relate to the target/output**, not about the data points themselves.
- Let me show you with examples.
- **Example 1:** Linear relationship
- **Scenario:** Predicting house price from house size.
- **Pattern:** If you double the size, the price doubles. Linear!



Size (sq ft)	Price (USD)
1000	200,000
2000	400,000
3000	600,000
4000	800,000

Non Linear Data

- **Scenario:** Predicting plant growth over time.
- **Pattern:** Height ≈ $(\text{days})^2$.
- Not linear, it's quadratic!
- A quadratic equation is a formula (like $y = x^2$).
- A parabola is the shape you get when you draw (graph) that quadratic equation.
- **Quadratic =** equation
- **Parabola =** graph (curve)
- **Analogy:** Song (equation) → Sound (graph)
- You sing a song → it produces sound
- You write a quadratic → it produces a parabola



Time (days)	Height (cm)
1	2
2	4
3	9
4	16
5	25

Real-world intuition

Linear relationships (common in simple scenarios)

- **Distance =** speed × time.
- **Total cost =** unit price × quantity.
- **Simple physics:** Force = mass × acceleration.

Non-linear relationships (very common in real problems)

- **Biology:** Population growth (exponential), enzyme kinetics (saturation curves).
- **Finance:** Compound interest ($A = P(1 + r)^t$), option pricing.
- **Vision:** A pixel's contribution to "is this a cat?" depends on its neighbors and context (highly non-linear).
- **Language:** Word meaning in context (depends on surrounding words in complex ways).

If Classical ML can Handle Nonlinear Patterns, then why do we Need Deep Learning?

ML also handles non-linearity

- Traditional ML can model non-linear relationships using different mechanisms:

A) Trees / Ensembles (Random Forest, Gradient Boosting)

- Learn **if/then splits** and interactions automatically.
- Very strong on **tabular/structured data**.
- Often win on business datasets with columns like age, income, category, etc.

B) Kernel methods Support Vector Machines (SVM) with a Radial Basis Function (RBF)

- Create non-linear decision boundaries via kernels.
- Good for some medium-sized datasets.

C) kNN

- Non-parametric; predicts from nearest examples.
- Can model complex boundaries but scales poorly.

Neural networks also model non-linearity, but the key difference is **how** they learn it and where they shine.

Main differences (practical + conceptual)

Feature engineering vs representation learning

- **Traditional ML (often):**
- Works great if you give it good features.
- You frequently do manual feature work:
  - transformations (log, square)
  - aggregations
  - domain-crafted signals
- **Neural networks / Deep learning:**
- Do **representation learning**: learn features automatically from raw inputs.
- Example:
  - Images: pixels → edges → shapes → objects
  - Text: tokens → syntax/semantics → meaning
- This is the single biggest conceptual difference.

Best Data Types (Where each Dominates)

Data type	Traditional ML (trees/boosting)	Deep learning
Tabular structured (columns)	Often best / strongest baseline	Sometimes good, often not best unless huge data
Images	Weak without hand-crafted features	Usually best (CNNs/ViTs)
Text / NLP	Weak with bag-of-words vs modern tasks	Usually best (Transformers)
Audio / speech	Limited	Usually best
Video	Limited	Usually best

A simple way to say it

- **Machine learning** = all learning models (linear, trees, SVM, neural nets, etc.).
- **Neural networks** = one ML approach that builds non-linear functions by stacking layers.
- **Deep learning** = neural networks with many layers that can learn features from raw data.

Manual Features Vs Automatic Features

Traditional ML pipeline (manual feature engineering)

- **Step 1 Input:** Raw image (pixels), e.g. 128×128×3.
- **Step 2 Manually design features** (human-chosen)
  - You don't feed raw pixels directly (or if you do, it performs poorly). You first extract engineered features such as:
- **HOG (Histogram of Oriented Gradients)**: Captures edge directions and local shapes (good for object outlines).
- **SIFT / SURF / ORB keypoints**: Detects distinctive points and describes local patches.
- **Color histograms**: How much of each color appears (sometimes useful, sometimes misleading).
- **Texture features (LBP, Gabor filters):** Captures fur-like texture patterns.

- After this, each image becomes a fixed feature vector:
- $x = \phi(image)$
- **Step 3:** Train a "classic" model: SVM, Logistic Regression, Random Forest, kNN
- **What's manual here?**

  You chose the feature extractor $\phi\left(\frac{\frac{HOG}{SIFT}}{etc}.\right)$ and its settings.
- **Limitation:**

  If the hand-designed features don't capture what matters (pose, lighting, background, breed variety), performance is limited.

Automatic Feature Learning

- **Step 1 Input:** Raw pixels (possibly resized/normalized).
- **Step 2 Neural network learns features automatically:** Use a CNN (e.g., ResNet) or Vision Transformer.
- The network learns a hierarchy:
- **Early layers:** edges, corners (similar to what HOG tries to capture)
- **Middle layers:** textures, patterns (fur, whiskers), simple shapes
- **Deeper layers:** parts (eyes, ears, snout)
- **Final layers:** whole object concept (cat vs dog)
- So the model learns:
- $\phi(\cdot)$ *automatically during training*
- **Step 3 Classifier head:** Final layer predicts cat/dog.
- **What's "manual" here?** Mostly basic preprocessing:
- resize/crop
- normalize pixel values
- maybe data augmentation (flip, rotate)
- But you did **not** manually define edges/textures/keypoints, those are learned.

Learning vs Inference: Conceptual Distinction

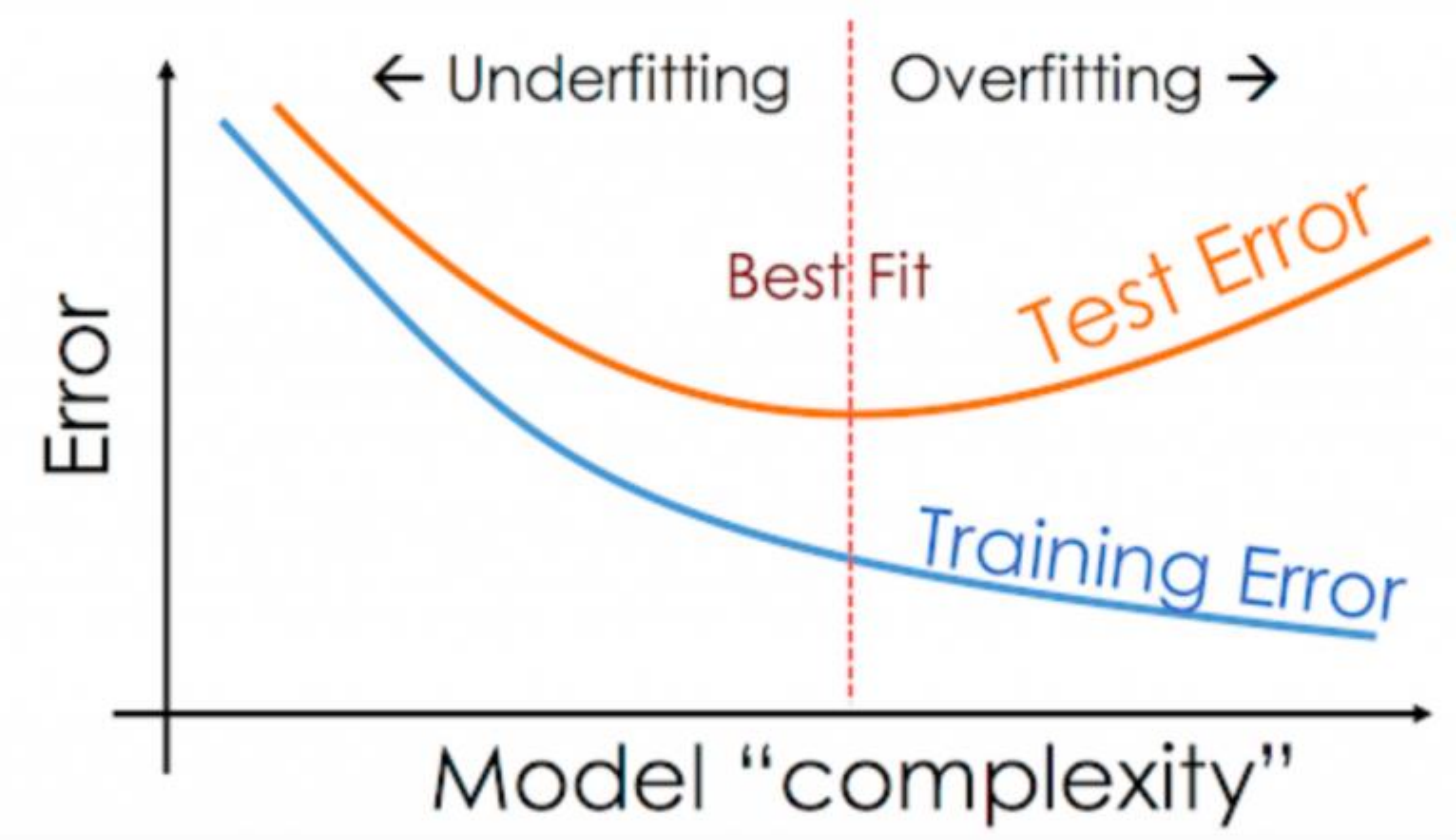
- **Learning (Training phase):**
- Model parameters are adjusted to fit training data
- Uses optimization algorithms (e.g., gradient descent)
- Computationally expensive, off-line (often on GPUs)
- **Inference (Prediction phase):**
- Using a trained model to make predictions on **new** inputs
- Usually faster, runs on servers or edge devices
- **Mental model:**
  - Learning = *studying for the exam*
  - Inference = *answering questions during the exam*

Generalization: The Central Goal

- **Generalization:**
- A model's ability to perform well on **unseen data** drawn from the same distribution as the training data.
- **Good learning:**
- Not just memorizing training data
- Captures underlying patterns/structure
- **Evaluation:**
- Train/validation/test splits
- Metrics (accuracy, error, loss)

Overfitting and Underfitting

- **Underfitting:**
  - Model is too simple to capture the data patterns
  - High error on both training and test data
- **Overfitting:**
  - Model fits noise in training data
  - Low training error; high test error
- **Just right (good fit):**
  - Low training error and low test error



Controlling Overfitting

- **Techniques to improve generalization:**
- More data / better data
- Regularization (e.g., weight decay, dropout)
- Early stopping based on validation performance
- Simpler architectures / fewer parameters
- Data augmentation (esp. in vision)
- **Important message:**
- Deep networks are powerful; **capacity** must be controlled carefully.

Types of Learning Paradigms

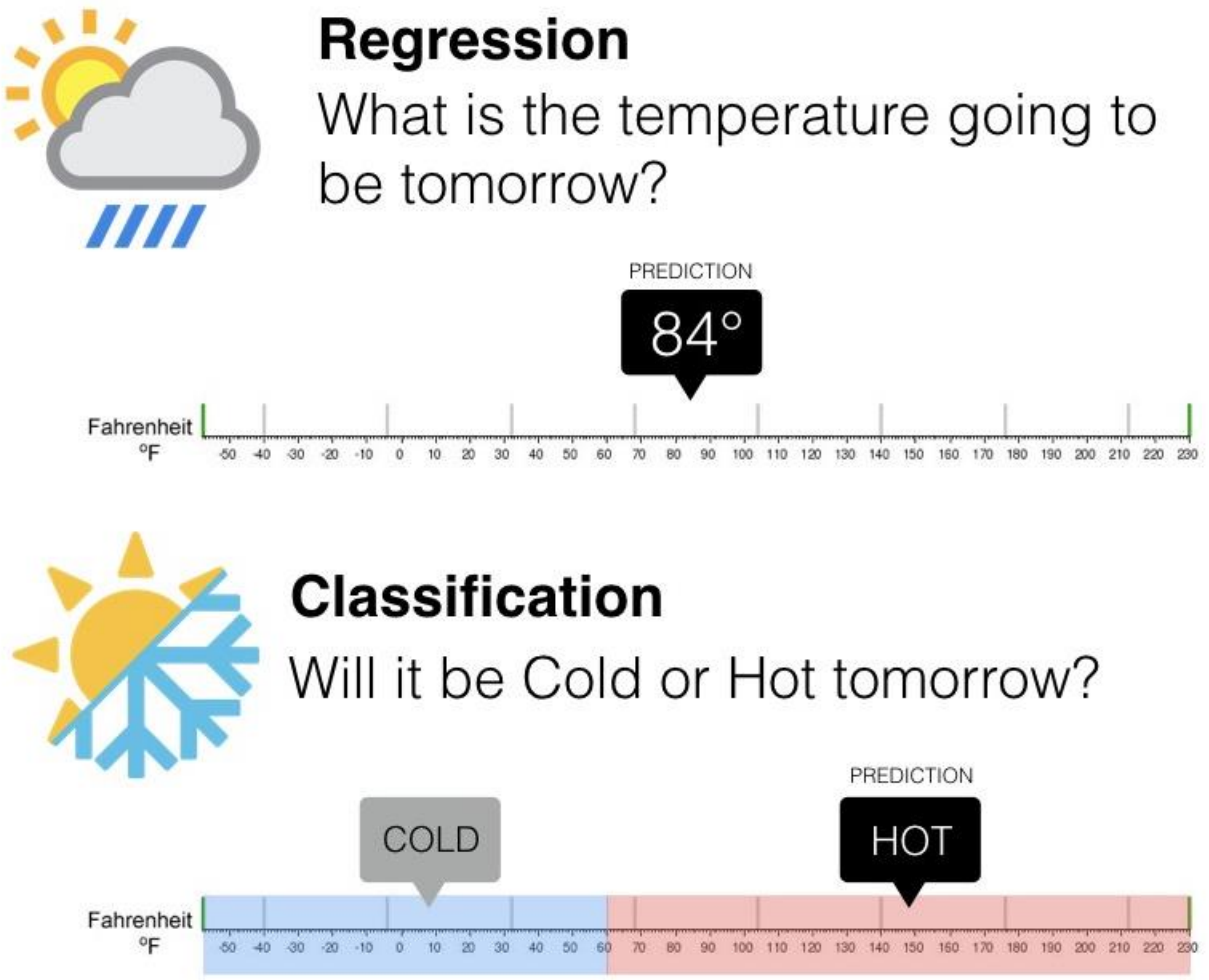
- Fully supervised learning
- Unsupervised learning
- Semi-supervised learning
- Self-supervised learning
- Reinforcement learning
- **Axis of difference:**
- Type and amount of **supervision / feedback** available.

Fully Supervised Learning

- **Setting:**
- You have input-output pairs: $\{(x_i, y_i)\}$
- Goal: learn a function f that maps x to y.
- **Examples:**
- Image classification (image → label)
- Sentiment analysis (text → sentiment)
- Speech recognition (audio → transcript)
- **Pros & cons:**
- **Pros:** Strong performance when labels are abundant
- **Cons:** Labelling is expensive/time-consuming

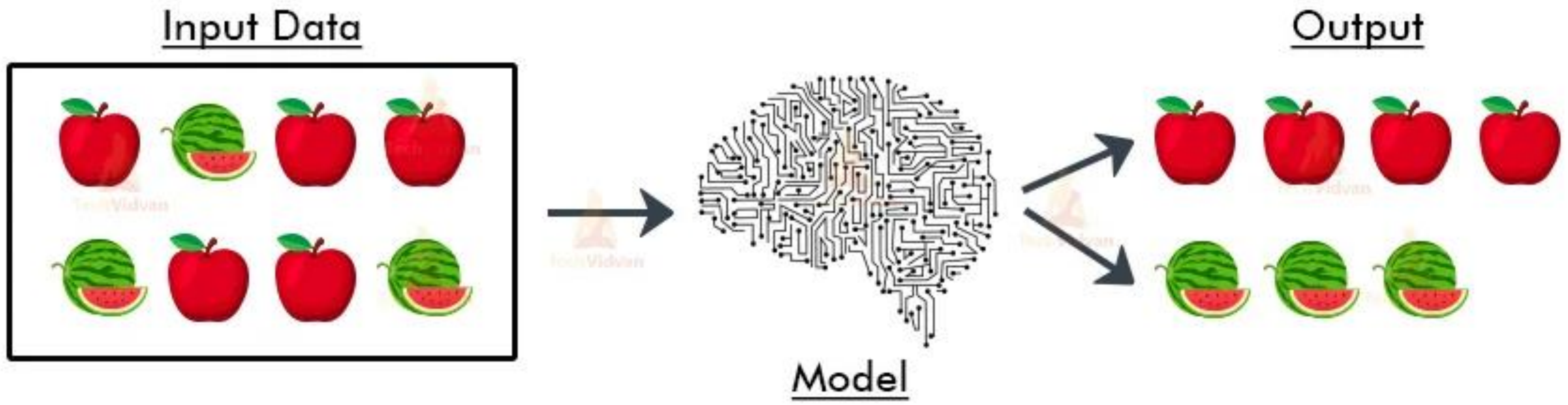
Regression vs Classification (Supervised)

- **Regression:** Predict **continuous** values
- House price prediction
- Stock price forecasting
- Temperature prediction
- **Common loss:** Mean Squared Error (MSE)
- **Classification:** Predict **discrete** labels
- Cat vs dog
- Spam vs not spam
- Multi-class object categories
- **Common loss:** Cross-entropy



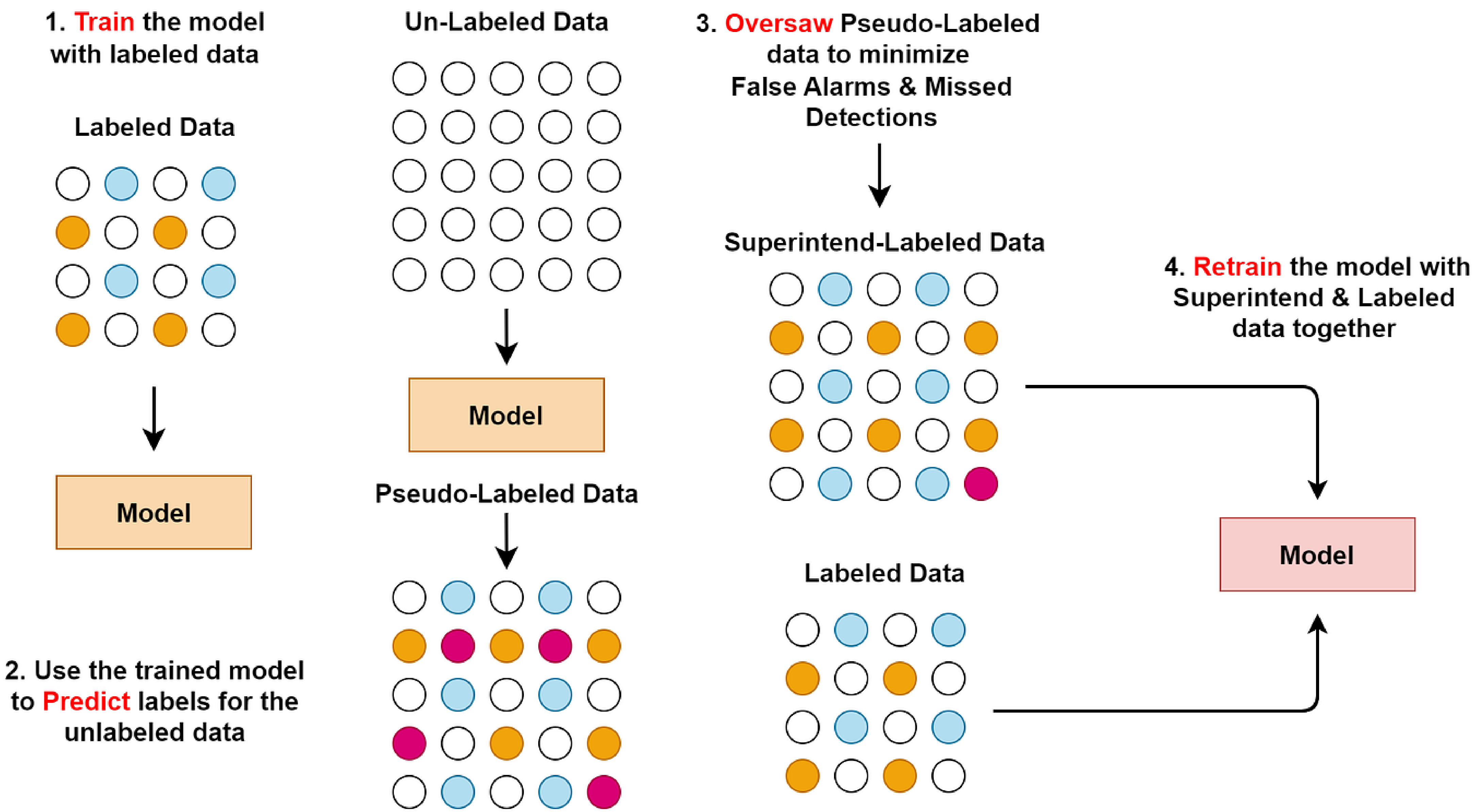
Unsupervised Learning

- Only input data $\{x_i\}$,**no labels**.
- Discover structure in data:
- Clustering (group similar examples)
- Dimensionality reduction (compress data)
- Density estimation, anomaly detection
- **Examples:**
- k-means clustering, PCA, autoencoders
- **Motivation:**
- Labels are scarce, but unlabeled data is cheap and abundant.



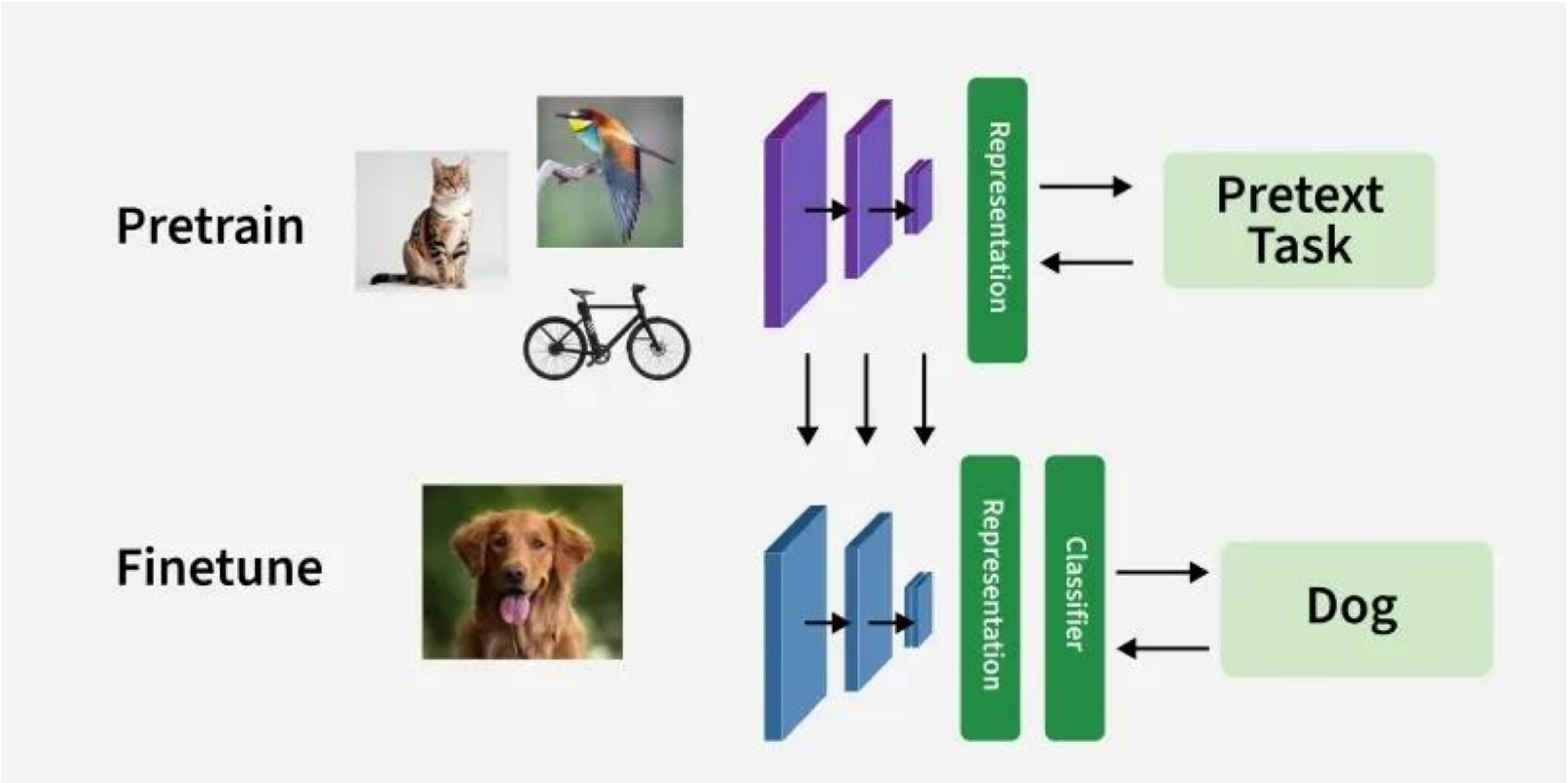
Semi-Supervised Learning

- Small amount of labeled data + large amount of unlabeled data
- Leverage unlabeled data to improve performance vs purely supervised learning with few labels
- **Approaches:**
- Consistency regularization
- Pseudo-labeling
- Graph-based methods
- **Use cases:**
- Medical imaging, low-resource languages, any domain where labeling is costly.



Self-Supervised Learning

- Create **pseudo-labels** from the data itself; solve pretext tasks.
- **Examples of pretext tasks:**
- Predict missing part of an image or sentence
- Next-token prediction in language
- Contrastive learning: distinguish true pairs vs mismatched pairs
- **Why important:**
- Enables learning powerful representations without manual labels
- Foundation of many modern **large language models** and vision models.



Reinforcement Learning: Delayed Feedback

- **Delayed Feedback:** You do many actions, but you get the reward **later**, not immediately.
- **Supervised Learning:**
- You give input → model predicts → you immediately give correct label.
  Instant feedback.
- **Reinforcement Learning:**
- Agent takes **many steps**, then gets reward at the end.
  No instant feedback.
- **Example**
- Agent walks 10 steps to reach the goal.
- It only gets the +10 reward **after reaching the goal**,
  NOT after each step.
- **This is delayed feedback.**

Reinforcement Learning : Sparse Feedback

- Rewards are rare.

  Most steps give **no reward**.
- **Supervised Learning:**
- Every example has a label.
- Feedback for every data point.
- **Reinforcement Learning:**
- Most steps = no reward.

  Only sometimes do you get reward.

- Example:
- Walk → 0 reward
- Walk → 0 reward
- Walk → 0 reward
- Reach goal → +10 reward
- **This is sparse feedback.**

Reinforcement Learning : Non-IID Data

- IID = Independent and Identically Distributed
- The data in RL is **dependent** on previous actions.

  You create your own data by moving in the environment.
- **Supervised Learning:**
- Each training example is separate and independent.

  Example:
- “I love this movie”
- “This is bad”
- “Amazing!”

  They don’t depend on each other.
- Data is IID.
- **Reinforcement Learning:**
- State 1 leads to State 2, which leads to State 3...

  Each next step depends on the previous step.
- Example: If agent moves right → next state is determined by that action.
- **Data is NOT independent.**

  This is non-IID.

What is Intelligence? Perspectives

- Defining Intelligence (Difficult Problem)
- **No single universally accepted definition.**
- **Informal views:**
  - Ability to learn, reason, adapt, and solve novel problems
  - Ability to achieve goals across a wide range of environments
- **Human vs Machine intelligence:**
  - Humans: embodied, emotional, social, grounded in physical world
  - Machines: currently narrow, task-specific, data-driven.

Perspectives on Intelligence

- **Psychological Perspective:**
  - Cognitive abilities: memory, reasoning, problem-solving, creativity
  - Measured historically via IQ tests (with many critiques)
- **AI perspective:**
  - Performance on benchmarks/tasks
  - Rational agents maximizing expected utility
- **Philosophical perspective:**
  - Consciousness, understanding, intentionality
  - "Strong AI" vs "Weak AI" debates

Is Deep Learning “Intelligent”?

- **Strengths:**
- High performance on specific tasks
- Can approximate complex functions, recognize patterns beyond human capability
- **Weaknesses:**
- Lack of robust common sense and causal reasoning
- Brittle outside training distribution
- Limited interpretability

- Deep learning systems exhibit *narrow* forms of intelligence; broader intelligence remains open research.

Real-World Applications Across Domains

Computer Vision

- **Tasks:**
  - Image classification, object detection, segmentation
  - Face recognition, activity recognition
- **Applications:**
  - Self-driving cars (perception stack)
  - Medical image analysis (tumor detection)
  - Surveillance and security
  - Industrial inspection and quality control

Natural Language Processing (NLP)

- **Tasks:**
  - Machine translation, summarization
  - Question answering, chatbots
  - Sentiment analysis, information extraction
- **Applications:**
  - Virtual assistants and customer support
  - Legal and financial document analysis
  - Search engines and recommendation systems

Speech, Audio & Multimodal Applications

- **Speech & audio:**
  - Speech recognition, text-to-speech
  - Speaker identification, emotion recognition
  - Music recommendation, sound event detection
- **Multimodal AI:**
  - Combining text, image, audio, video
  - Examples: video captioning, visual Q&A, AR/VR assistants

Robotics and Control

- **Applications:**
  - Industrial robots and cobots
  - Drones and autonomous vehicles
  - Household robots (vacuum cleaners, assistive devices)
- **Role of deep learning:**
  - Perception (seeing the world)
  - Policy learning (RL) for complex control tasks
  - Sensor fusion and decision-making.

Finance, Business & Recommendation

- **Finance:**
  - Algorithmic trading, risk prediction, fraud detection
- **Business:**
  - Demand forecasting, inventory optimization
  - Customer segmentation, churn prediction
- **Recommender systems:**
  - E-commerce product recommendations
  - Personalized content feeds
- **Value proposition:**
  - Data-driven decisions, improved efficiency, personalization.

Science & Medicine

- **Scientific discovery:**
  - Protein structure prediction
  - Materials discovery; drug design
  - Physics simulations and surrogates
- **Medicine:**
  - Predictive models for disease risk
  - Personalized treatment recommendations
  - Clinical decision support systems

Paper Writing

- A typical paper must consist of following main section
  - Abstract
    a. Should be one paragraph explaining what is being presented in the paper
  - Introduction
    a. A short explanation of the problem and reviews of the existing research.
    b. You must download and review at least 10-15 papers (Journal Papers are highly encouraged)
  - Data Set
    a. Which data has been used for this study
  - Methodology
    a. What method(s) has been implemented
  - Experimental Results
    a. Explanation of the results obtained. Should include figures, tables etc.
  - Conclusion
    a. Summarize what you did overall
  - References
    a. Bibliography of which papers were reviewed or used in the paper

Getting Started with Research

- Step 1. Select a Topic
  - Your interests and Skills
  - Supervisor / Project
  - Availability of resources
- Step 2. Search for Relevant Information
  - IEEE Explore
  - Springer
  - Scopus
  - Web of Science
  - Google Scholar
- Step 3. Organize Resources
  - Zotero
  - EndNote
  - Mendeley
  - JabRef
  - License
  - MarginNote
- Step 4. Write Paper
  - Outline the Paper
  - Take Notes
  - Avoid Plagiarism

Your Tasks

I will review this in the same day next class and assign the corresponding marks.