Introduction to Machine Learning and Artificial Intelligence

Gustavo Martin Larrea Gallegos

April 29, 2025

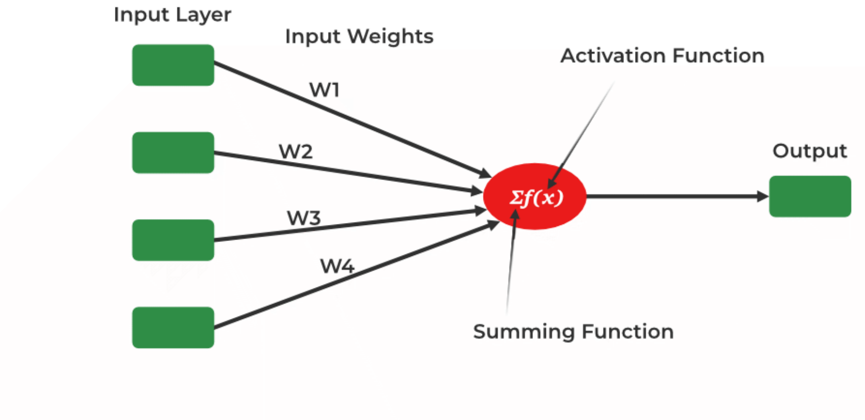
What is Machine Learning?

- ML is the set of ‘algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions‘
- ML is the ‘study of computer algorithms that improve automatically through experience‘

History of Machine Learning - Early Seeds (Pre-1960s)

- **1950: Turing Test** - Proposed by Alan Turing as a test of a machine's ability to exhibit intelligent behavior.
- **1956: Dartmouth Workshop** - John McCarthy coins the term **Artificial Intelligence**. Logic Theorist program presented (Newell, Simon, Shaw).
- **1956: Arthur Samuel's Checkers Player** - Demonstrated **learning from experience**, popularizing "Machine Learning" (1959 paper). Learned parameters and used search.
- **1957-58: Perceptron** - Frank Rosenblatt develops the first **artificial neural network** based on McCullach-Pitts neuron model and Hebbian learning.

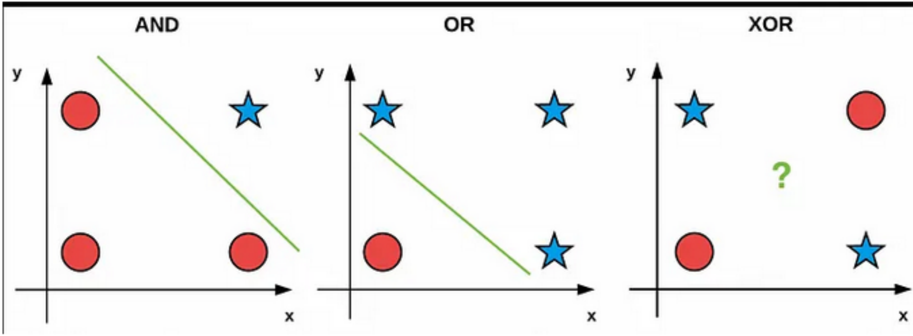
History of Machine Learning - Early Seeds (Pre-1960s)



History of Machine Learning - Growth & Challenges (1960s-1980s)

- **1963: Support Vector Machines (SVMs)** - Early theoretical work by Vapnik & Chervonenkis (popularized later).
- **1969: Perceptron Limitations** - Minsky & Pappert show single-layer perceptrons cannot solve XOR, contributing to the first **"AI Winter"**.
- **1970s-Early 80s: Symbolic AI Dominates** - Rule-based expert systems become the focus. Less emphasis on learning from data.

History of Machine Learning - Early Seeds (Pre-1960s)



History of Machine Learning - Revival & Modern Era (Mid 1980s - Present)

- **1980s: Bayesian Networks** - Judea Pearl introduces probabilistic graphical models for reasoning under uncertainty.
- **1986: Backpropagation Rediscovered** - Rumelhart, Hinton, & Williams popularize an efficient method for training multi-layer neural networks.
- **1980s-1990s: Shift Back to Learning** - Expert systems decline; ML gains traction.
- **1997: Deep Blue** - IBM's computer defeats world chess champion Garry Kasparov.
- **2012: AlexNet & Deep Learning Boom** - AlexNet wins ImageNet, ushering in the era of **deep learning**.
- **2012 - Present: "AI Spring"** - Rapid advances, super-human performance in many tasks (e.g., AlphaGo 2017).

Basic Terminology - Input & Output

X (Input):

- Features
- Predictors
- Independent Variable
- Covariate

Y (Output):

- Target
- Label (especially in classification)
- Prediction (the model's guess for Y)
- Dependent Variable
- Response Variable

ML Algorithm / Model / Hypothesis: The function that maps X to Y.

Basic Terminology - Data

- **Instance / Example:** A single data point, often (x, y) or just x.
    - *Example Input x:* `<tumorsize=18.2, texture=27.6, ...>`
- **Dataset:** A collection of instances.
    - *Example Structure:* $D = \{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$
- **Features:** The individual measurable properties used as input (components of x).
- **Labels / Targets:** The known correct outputs (y) for training instances in supervised learning.

Basic Terminology - Process

- **Training Data:** Subset used to *build* or *train* the model.
- **Test Data:** Subset held back to *evaluate* performance on unseen data.
  - *Crucial:* Model should NOT see test labels during training.
- **Ground Truth:** The true labels/values in the test set for comparison.
- **Evaluation:** Measuring performance by comparing predictions against ground truth.

Families of Machine Learning Methods - Overview

Algorithms are broadly categorized based on data and problem type:

1. **Supervised Learning**
2. **Unsupervised Learning** (& Self-supervised)
3. **Semi-supervised Learning**
4. **Reinforcement Learning**

Families of ML - 1. Supervised Learning

- **Goal:** Learn a mapping from input X to output Y.
- **Data:** Requires **labeled** data $D = \{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$. Learns from examples with known answers.
- **Tasks:**
  - **Classification:** Output Y is a discrete category/class (e.g., spam/not spam, object class).
  - **Regression:** Output Y is a continuous value (e.g., house price, temperature).
- Dominant approach for many practical applications.

Families of ML - 2. Unsupervised Learning

- **Goal:** Discover patterns, structure, or representations in input data X *without* explicit labels Y.
- **Data:** Uses **unlabeled** data $D = \{x^{(n)}\}_{n=1}^{N}$.
- **Tasks:**
  - **Clustering:** Group similar instances.
  - **Dimensionality Reduction:** Reduce number of features.
  - **Density Estimation / Generative Modeling:** Learn data distribution (e.g., generate images).
  - **Anomaly Detection:** Identify unusual data points.
- **Self-supervised Learning:** Create proxy supervised tasks *from* unlabeled data to learn features (e.g., predict masked words).

- **Goal:** Learn when *some* instances are labeled, but most are unlabeled.
- **Data:** A mix of labeled (x, y) pairs and unlabeled x instances.
- **Idea:** Leverage abundant unlabeled data to improve learning from the small labeled set.
- **Examples:**
  - Website classification with few manual labels.
  - Recommendation systems (Matrix Completion) with sparse ratings.

Families of ML - 4. Reinforcement Learning (RL)

- **Goal:** Learn how an **agent** takes **actions** in an **environment** to maximize cumulative **reward**.
- **Data:** No predefined dataset; learns via trial-and-error interaction. Receives (state, reward) feedback.
- **Characteristics:** Sequential decision making, delayed rewards, exploration vs. exploitation.
- **Examples:** Game playing (Atari, Go), robotics control.
- **Related:** Imitation Learning (learning from demonstrations).

Summary

- **History:** ML evolved from early AI, through challenges, to the current deep learning era.
- **Terminology:** Understanding Input (X), Output (Y), Instances, Datasets, Features, Labels, Train/Test is crucial.
- **Families:**
  - **Supervised:** Labeled data (Classification/Regression).
  - **Unsupervised:** Unlabeled data (Clustering, Generative, Self-supervised).
  - **Semi-supervised:** Mix of labeled/unlabeled data.
  - **Reinforcement:** Learns via rewards through interaction.

What are Large Language Models (LLMs)?

- LLMs are a type of Artificial Intelligence (AI) model.
- They are based on deep learning, specifically large neural networks.
- **Key Characteristic:** Trained on massive amounts of text data (like books, articles, websites). [1, 6]
- **Goal:** To understand and generate human-like text. [1]
- They learn grammar, facts, reasoning abilities, and even biases from the data they are trained on. [6]
- Think of them as incredibly advanced auto-complete systems that can do much more.

How do LLMs work? (The Basics)

- At their core, many LLMs work by predicting the next word in a sequence.
- Given an input (a "prompt"), they generate text word by word, based on patterns learned during training.
- **Underlying Technology:** Most modern LLMs use an architecture called the "Transformer". [6]
- This architecture allows the model to weigh the importance of different words in the input sequence when generating the output.
- Training involves adjusting millions or billions of internal 'knobs' (parameters) to minimize prediction errors on the training data. [1]

What can LLMs do?

LLMs are versatile and used in many applications:

- **Text Generation:** Writing essays, poems, code, emails.
- **Translation:** Translating text between languages.
- **Summarization:** Condensing long documents into key points.
- **Question Answering:** Answering questions based on the knowledge learned during training or provided context. [1]
- **Chatbots & Virtual Assistants:** Powering conversational AI like ChatGPT. [1]
- **Code Generation:** Assisting programmers by writing or debugging code.

Model Size: The "Large" in LLMs

- The size of an LLM is typically measured by the number of **parameters** it has. [1, 4]
- Parameters are the internal variables (weights and biases) the model learns during training. [1]
- Think of parameters as the 'knowledge capacity' of the model.
- Sizes range widely: [4]
  - **Smaller Models:** Millions to a few billion parameters (e.g., BERT-Large - 340M, GPT-2 - 1.5B).
  - **Large Models:** Tens to hundreds of billions of parameters (e.g., GPT-3 - 175B, BLOOM - 176B, Llama 2 - 7B to 70B). [4]
  - **State-of-the-Art Models:** Can have hundreds of billions or even trillions of parameters (e.g., GPT-4 - parameter count not officially disclosed but estimated to be very large). [4]

Why Does Model Size Matter?

- **Capability:** Generally, larger models (with more parameters) tend to be more capable, understand nuances better, and perform more complex tasks. [1]
- **Data Needs:** Larger models require significantly more data for effective training.
- **Computational Cost:** Training and running larger models require exponentially more computational power and memory. [1, 5]
- **Accessibility:** Very large models are expensive to train and often require specialized hardware to run efficiently. [5, 7]

Hardware Requirements: Training LLMs

Training large LLMs from scratch is extremely resource-intensive:

- **GPUs are Essential:** Requires massive clusters of high-end GPUs (Graphics Processing Units), often hundreds or thousands. [5, 7] Examples: NVIDIA A100s or H100s. [5]
- **Vast Memory (RAM & VRAM):** Both system RAM and GPU memory (VRAM) are critical to hold the model parameters and training data batches. [7] Terabytes of RAM might be needed.
- **High-Speed Interconnects:** Fast connections between GPUs and nodes (like NVLink, InfiniBand) are crucial for distributed training. [5]
- **Storage:** Petabytes of storage for the enormous datasets.
- **Cost:** Training state-of-the-art models can cost millions of dollars in hardware and electricity. [7]
- **Feasibility:** Essentially impossible for individuals or most university departments; typically done by large tech companies or research consortia. [5]

Hardware Requirements: Running LLMs (Inference)

Running a pre-trained LLM (inference) is less demanding than training, but still significant:

- **Hardware Depends on Model Size:** Smaller models might run on consumer hardware, while larger ones need powerful servers. [7]
- **GPU Importance:** GPUs significantly speed up inference. [7] The amount of VRAM on the GPU is often the main bottleneck – it needs to be large enough to hold the model's parameters. [2, 7]
- **RAM:** Sufficient system RAM is also needed to load the model and handle data. [7]
- **CPU:** A decent CPU is required, but the GPU does most of the heavy lifting for LLM computations. [7] CPU-only inference is possible but often very slow for larger models. [3]
- **Quantization:** Techniques exist to shrink models (quantization) to run on less powerful hardware, sometimes with a trade-off in accuracy. [2]

What Hardware Can Students Use?

Getting hands-on experience:

- **Cloud Platforms:**
  - **Google Colab:** Offers free (with limitations) and paid access to GPUs (like NVIDIA T4 or V100). Great for experimenting with smaller models or tutorials. [2]
  - **Kaggle Kernels:** Similar to Colab, provides free GPU access.
  - **Cloud Providers (AWS, Azure, GCP):** Offer more powerful GPU instances, but can be costly (student credits might be available).
- **Local Machines (Your PC/Laptop):**
  - **High-End Consumer GPUs:** Modern NVIDIA GPUs (e.g., RTX 30xx, 40xx series) with substantial VRAM (8GB minimum, 12GB+ recommended) can run many medium-sized models locally. [2, 3]
  - **CPU Inference:** Possible for smaller models using frameworks like Llama.cpp, but expect slow generation speeds. [3]
  - **RAM:** 16GB is often a minimum, 32GB+ recommended for running larger models locally.
- **University Resources:** Check if your university provides access to HPC (High-Performance Computing) clusters with GPUs.

Key Takeaways

- LLMs are powerful AI models trained on vast text data to understand and generate language. [1, 6]
- Model size (parameters) is a key factor influencing capability and resource needs. [1, 4]
- Training large LLMs requires massive, expensive computational resources (GPU clusters). [5, 7]
- Running LLMs (inference) is less demanding but still requires significant resources, especially VRAM for larger models. [2, 7]
- Students can experiment with LLMs using cloud platforms (Colab), powerful personal computers with GPUs, or university resources. [2, 3]

- These slides are adapted from the material provided at the Applied Machine Learning course at McGill University (https://www.cs.mcgill.ca/ isabeau/COMP551/F23/index.html).