Introduction to
AI Safety, Ethics, and Society

Artificial Intelligence Fundamentals

Roadmap

1. Artificial Intelligence: *Definitions, history, and types.*

1. Machine Learning: *Models, tasks, data, and learning types.*

2. Deep Learning: *The preeminent paradigm.*

1. Scaling Laws: *Many AI systems change predictably with scale.*

1. Speed of AI Development: *xxx*

Artificial Intelligence

Definitions, history, and types

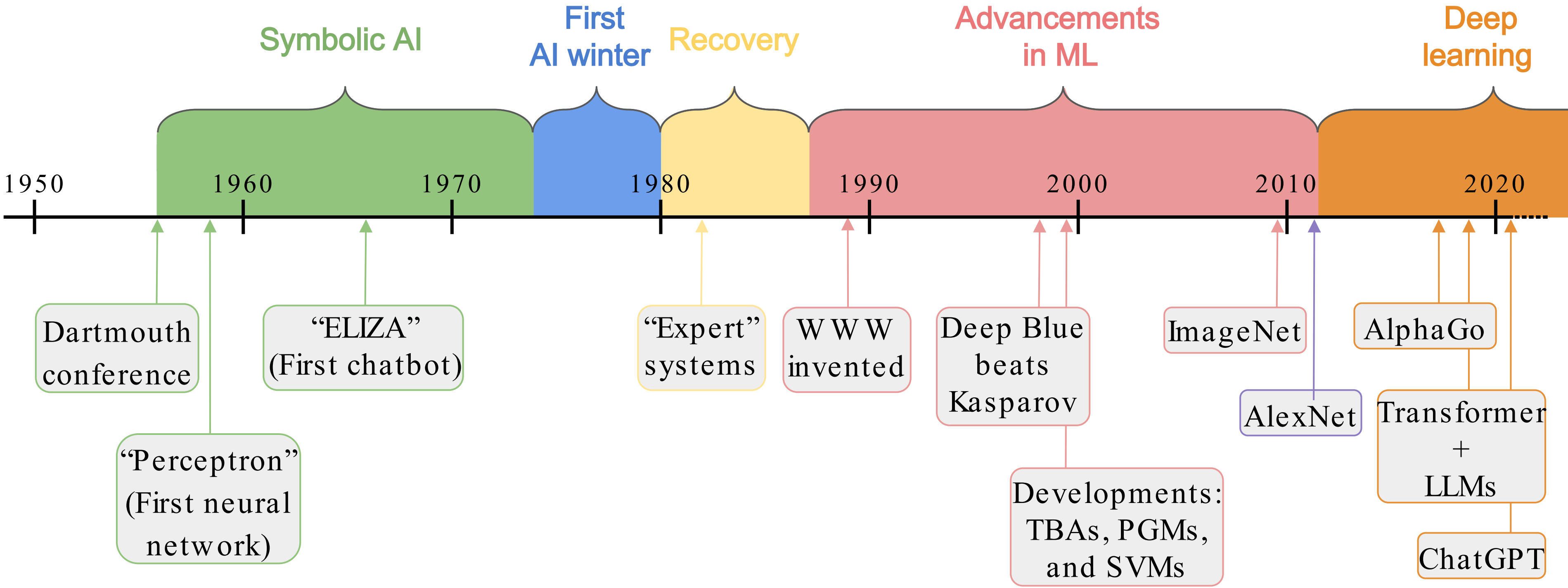
Definition of AI

There is no single universally-accepted definition of "artificial intelligence." It is used to describe different (but related) ideas:

- Branch of computer science
- Type of machine
- Tool
- Component of a business model
- Philosophical idea

Our definition: computer systems that perform tasks typically associated with intelligence such as problem solving, decision making, forecasting.

Modern History of AI



Types of AI

Narrow AI : capable of performing specific tasks only.

Artificial General Intelligence (AGI) : capable of performing many cognitive tasks across multiple domains, at human or superhuman levels.

Human-level AI (HLAI) : capable of doing virtually everything humans do.

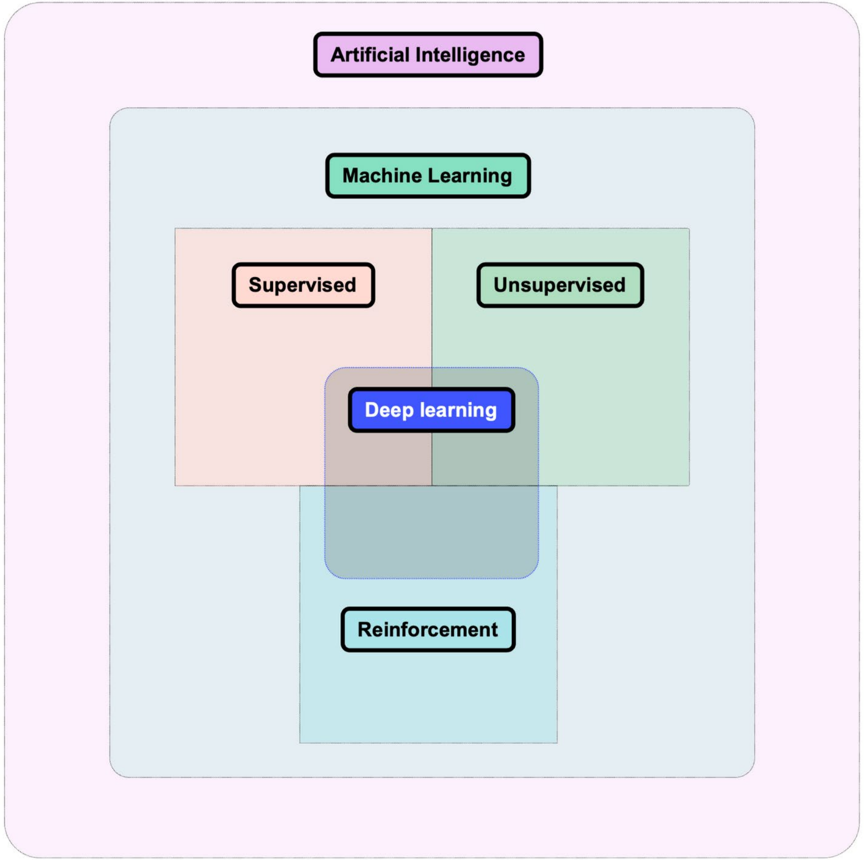
Transformative AI (TAI) : systems with a dramatic impact on the world at the level of the industrial revolution.

Superintelligence (ASI) : capable of outclassing human performance on virtually all intellectual tasks.

Machine Learning

The models, tasks, data, and types of learning

Machine Learning Definitions



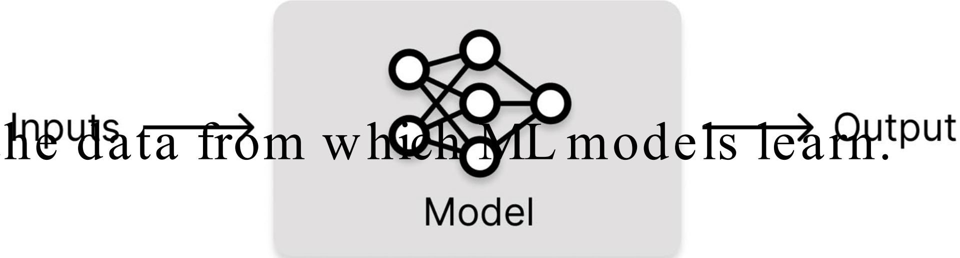
Machine Learning Definitions

Machine learning (ML) : developing computer systems that can learn directly from data without following explicit pre-set instructions.

Algorithm : a procedure (recipe) for accomplishing a task.

ML model : an algorithm designed to learn automatically from data by identifying patterns and relationships to make predictions or decisions.

Training data : the data from which ML models learn.



Model Access

Model developers can grant different levels of **access** to the public.

Closed Models: Operate in a closed system, not accessible for external use or modification, often used internally by organizations.

API Models: Accessible through an interface with usage restrictions, allowing users to query the model without viewing or modifying the underlying model.

Open Models: Fully accessible and modifiable, often part of open-source projects, allowing users to use, modify, and distribute freely.

Computational Resources

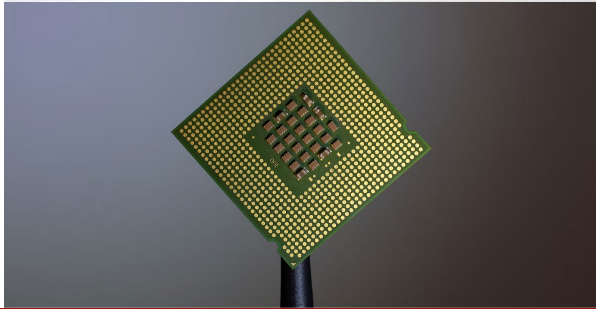
Compute in ML: Essential for handling the large datasets and complex calculations in machine learning, with GPUs and CPUs being primary examples.

GPUs (Graphics Processing Units) :

- Parallel processing

CPUs (Central Processing Units) :

- Sequential processing



Tasks, Inputs, Outputs

1. **Task**: What is the primary goal of the ML model?

1. **Inputs** : What data does the ML system receive?

1. **Outputs** : What does the ML system produce?

Types of Input Data

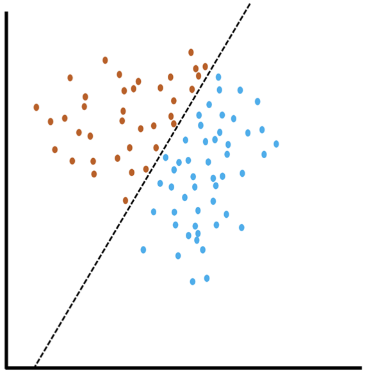
Modality : a type of input to an ML model

Major modalities in machine learning:

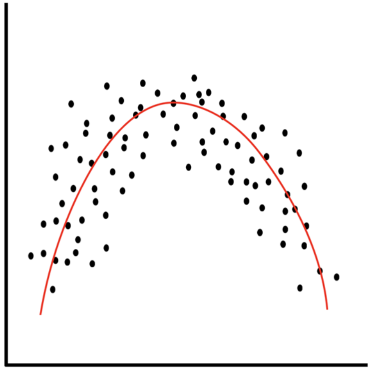
- Tables
- Text
- Images
- Video
- Audio
- Time series
- Graphs
- Set values

Key ML Tasks

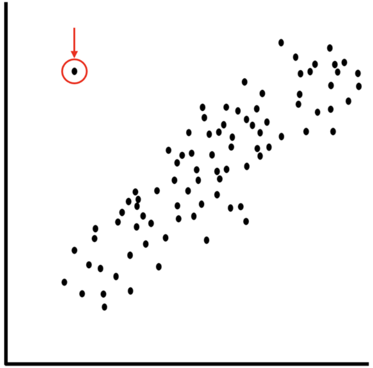
Classification



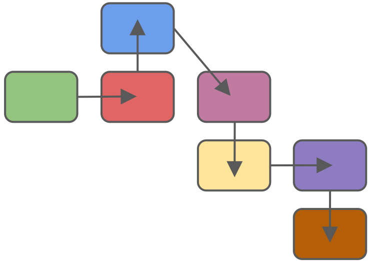
Regression



Anomaly detection



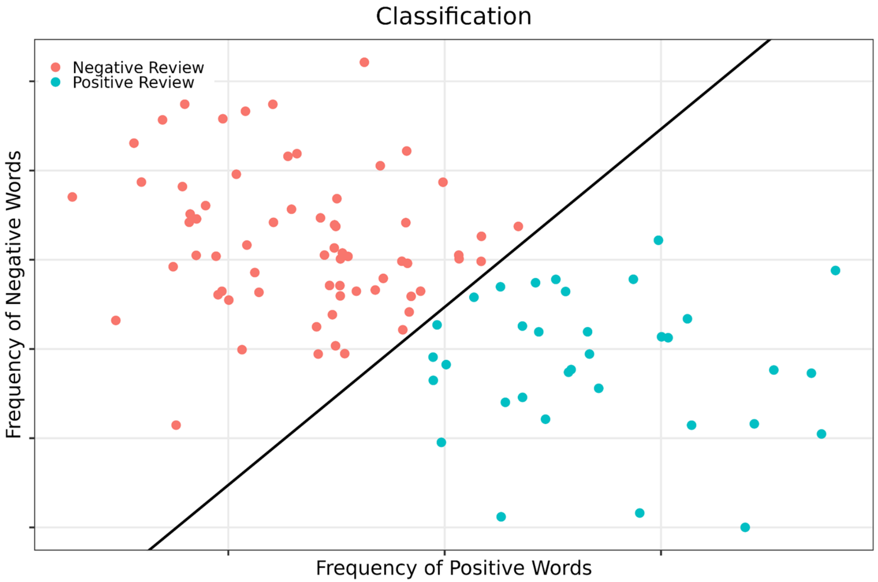
Sequence modelling



Task 1:

Classification

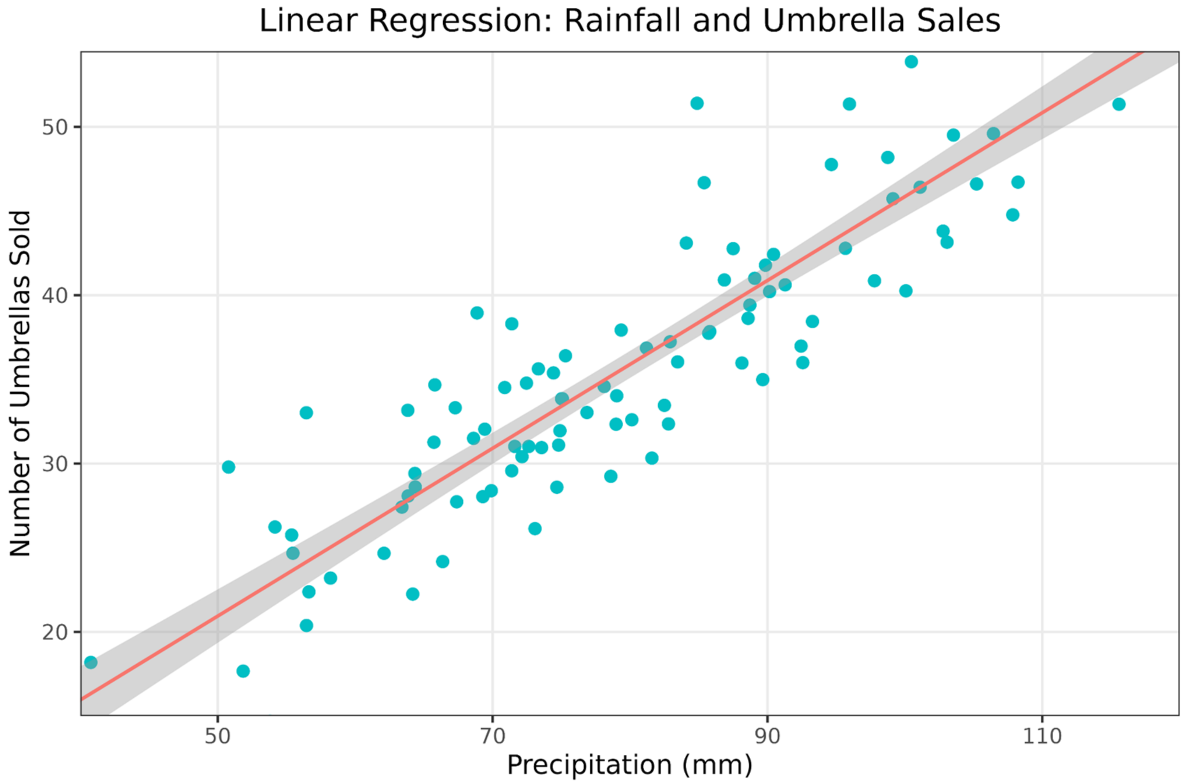
Classification : predicting categories or “classes” using features of an input data point.



Task 2:

Regression

Regression: predicting numerical outputs using features of input data.



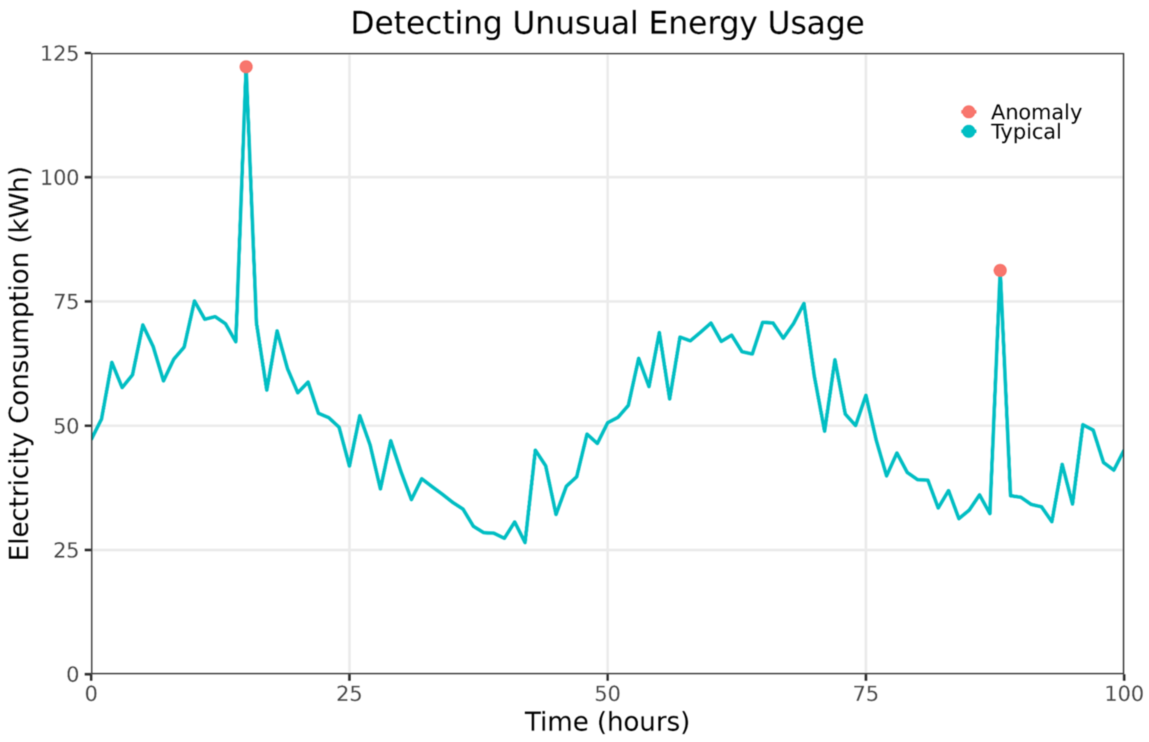
Task 3:

Anomaly Detection

Anomaly detection : identifying outliers or abnormal data points.

Example: black swan detection.

Black swans: events that are rare and unpredictable, yet impactful



Task 4:

Sequence Modeling

Sequence modeling : analyzing and predicting patterns in sequential data.

Generative modeling : creating new data that resembles the input data; ChatGPT is a “generative model”

Components of the ML Pipeline

ML pipeline : a series of interconnected steps in the process of developing a ML model.

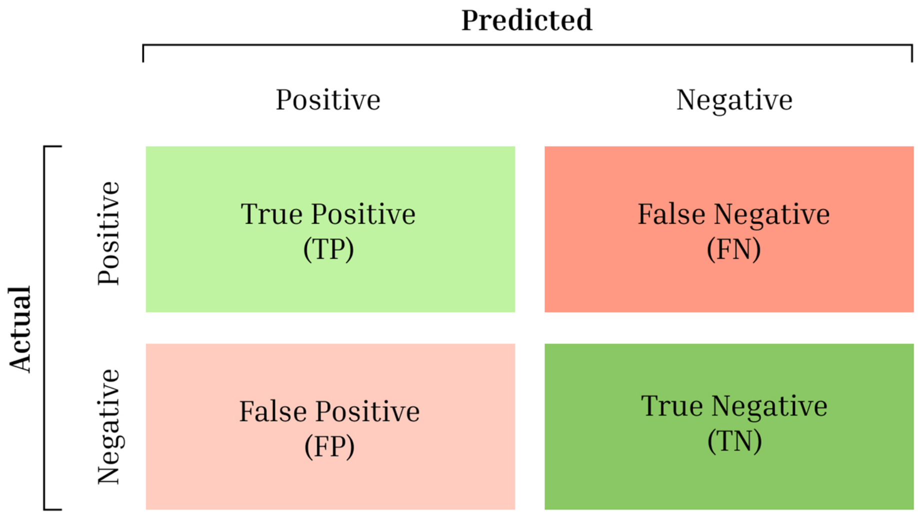
Steps:

- Data collection
- Labeling
- Choosing architecture
- Training
- **Evaluation**
- Deployment

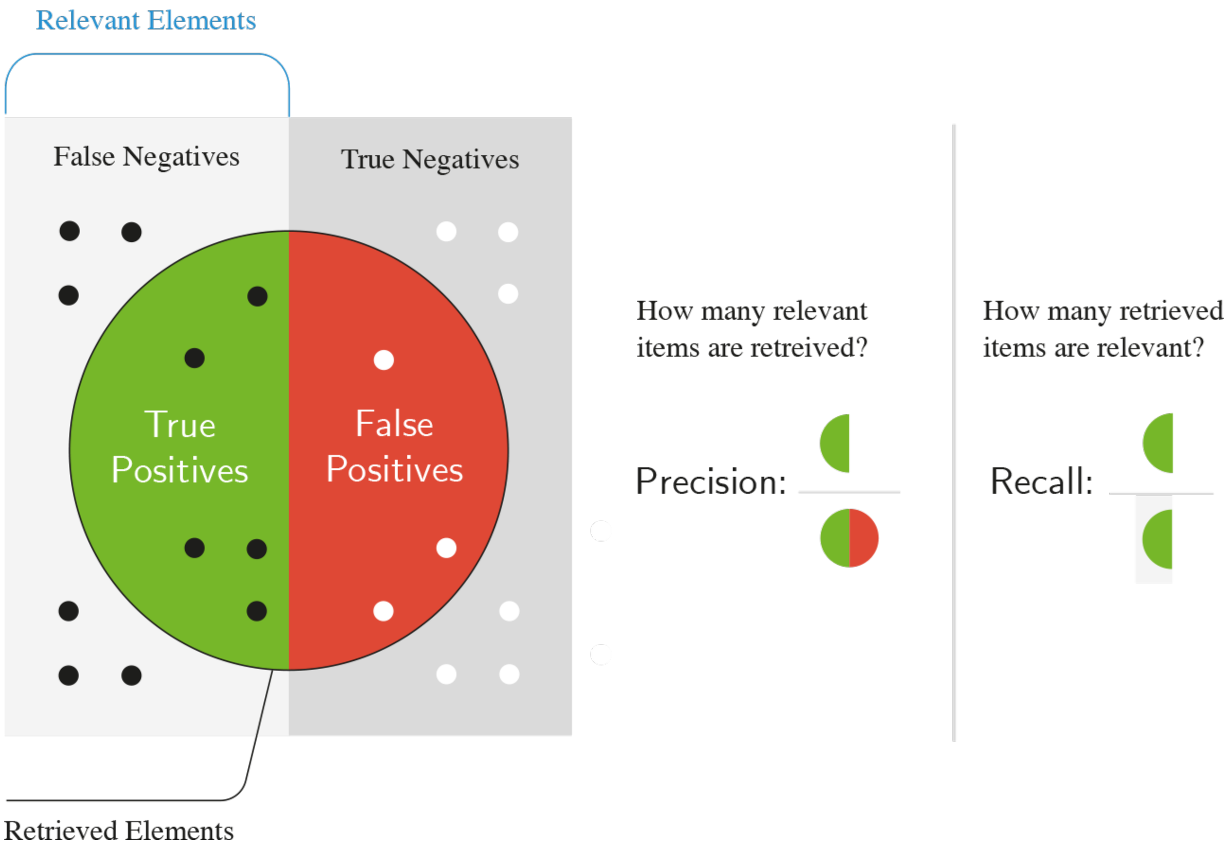
Evaluating ML Models (1/3)

Metrics:

- **Accuracy**: percentage of correct predictions
- **Precision**: the purity of the flagged examples
- **Recall**: proportion of relevant examples that were flagged



Evaluating ML Models (2/3)



Evaluating ML Models (3/3)

We can have many goals:

- **Predictive power** : the amount of error in predictions
- **Inference/latency** : the speed of result production
- **Transparency** : ability to understand workings producing output
- **Reliability** : performance consistency (over time/in varied conditions)
- **Scalability** : capacity to maintain/improve performance as we scale a *key variable*:
  - Compute
  - Parameter count
  - Dataset size

Three Types of Machine Learning

Supervised Learning

Unsupervised Learning

Reinforcement Learning

Type 1:

Supervised Learning

Supervised learning : learning from labeled data.

- Learning relationships between input data and output labels
- Labels are usually generated by humans

Example: Species identification

Advantage: high accuracy and reliability for large and well-labeled dataset.

Disadvantage: worse performance for loosely-defined tasks, like poetry or image generation; may also require manual labeling.

Type 2:

Unsupervised Learning

Unsupervised learning : learning from unlabeled data.

- Discovering and identifying patterns within a dataset to learn the relationships between variables
- Also known as “self-supervised learning,” because the model may use these patterns to generate supervisory signals for itself

Examples:

- Language models
- Image inpainting

Type 2:

Unsupervised Learning

Type 3:

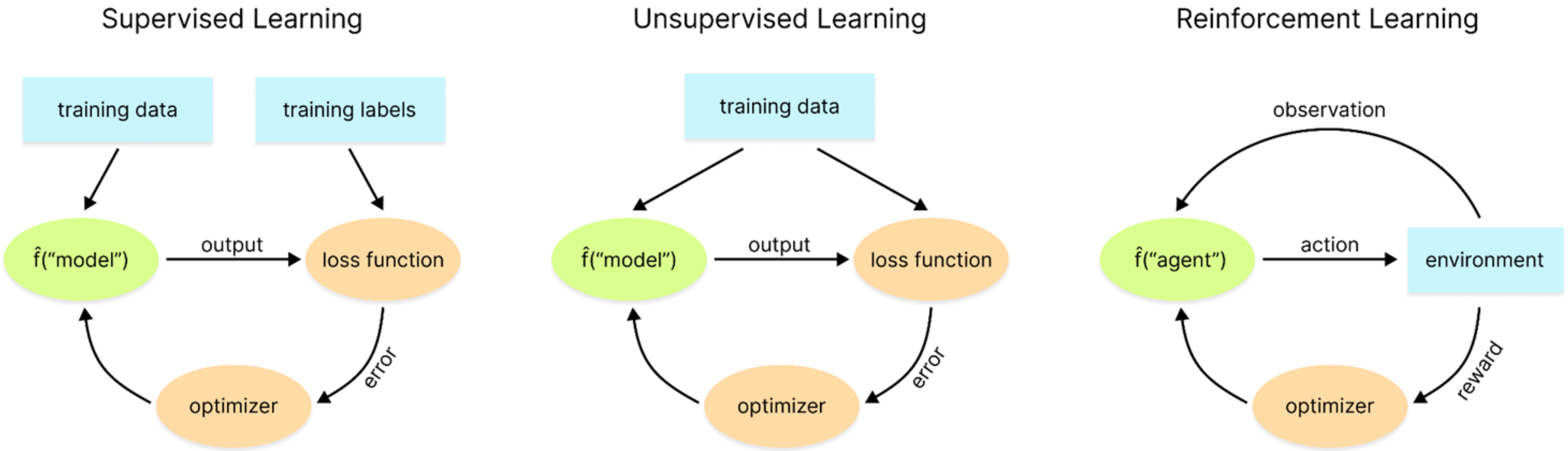
Reinforcement Learning

Reinforcement learning (RL) : learning from agent-gathered data.

- AI agents making decisions and improving their performance based on responses to their environment

Example: robots navigating environments through trial-and-error.

Review of Machine Learning Types



Recap

AIs are computer systems that perform tasks typically associated with intelligence such as problem solving, decision making, forecasting. They can be classified according to their capabilities.

Machine Learning systems can learn directly from data without following explicit pre-set instructions. They are used for tasks like classification, regression, anomaly detection, and sequence modeling.

ML requires data, architecture, training it, evaluating, deploying.

ML can use supervised, unsupervised, and reinforcement learning.

Roadmap

1. Artificial Intelligence: *Definitions, history, and types.*

1. Machine Learning: *Models, tasks, data, and learning types.*

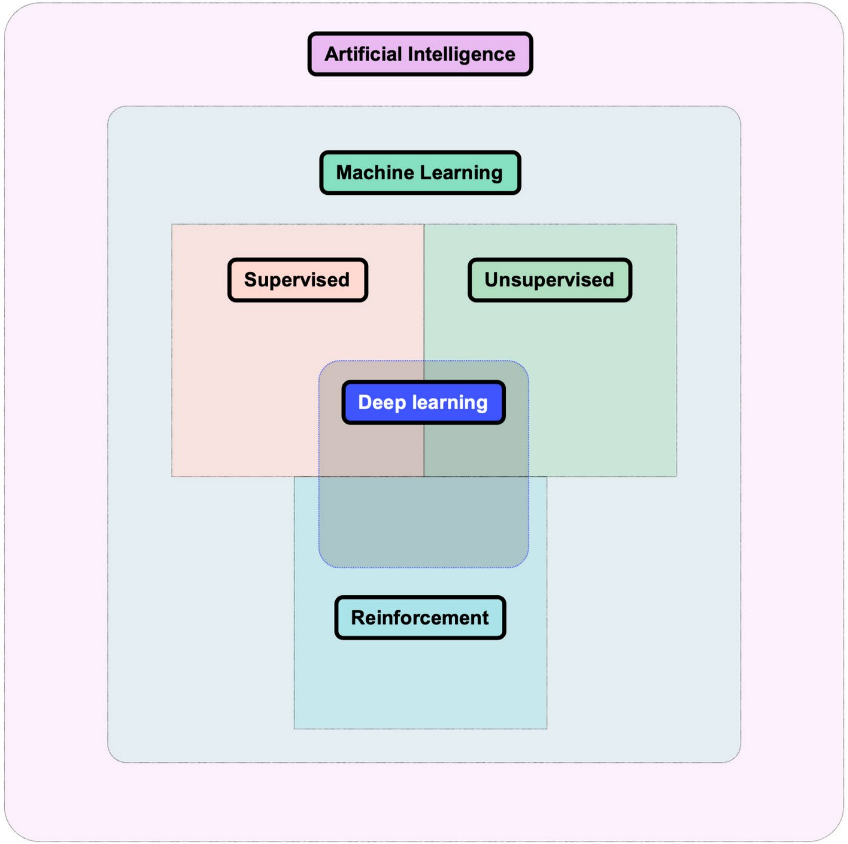
2. Deep Learning: *The preeminent paradigm.*

1. Scaling Laws: *Many AI systems improve predictably with scale.*

Deep Learning

The preeminent paradigm

Introduction to Deep Learning



Introduction to Deep Learning

Deep learning (DL) :

- Uses **neural networks** : layers of interconnected nodes that transform inputs into outputs, inspired by biological neurons.
- Extracts useful patterns from large quantities of data
- Useful for learning hierarchical representations of the world

Why Deep Learning Matters

Performance:

- Superhuman in some domains (though not reliably so)
- Examples: image recognition, chess, and Go

Real-world usefulness:

- Wide variety of useful applications
- Examples: healthcare, social media, and autonomous vehicles

Scalability:

- DL models are highly scalable
- Performance tends to improve with more data, rather than tapering off

Three Factors of DL Model

DL model performance is affected by:

- Algorithms
- Data
- Compute

DL Model Building Blocks

1. Multi-layer perceptrons (MLPs)
2. Activation functions
3. Residual connections
4. Convolution
5. Self-attention

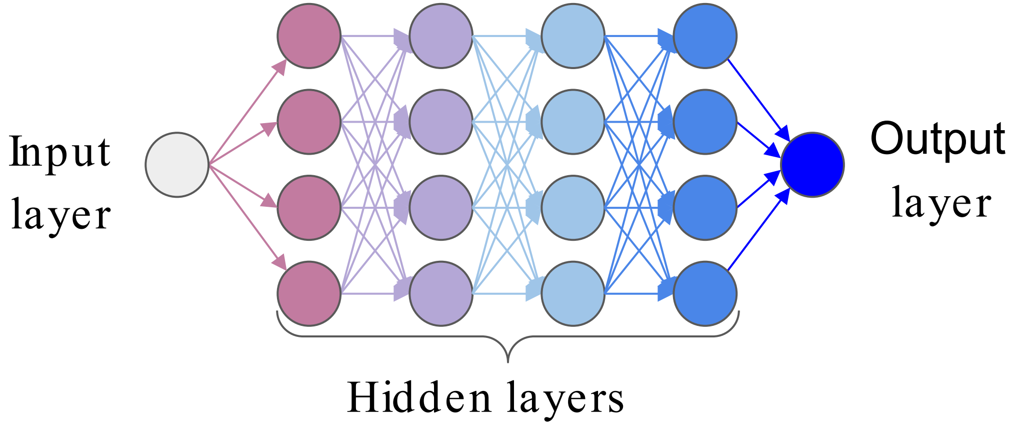
Building block 1:

Multi-Layer Perceptrons

Multi-layer perceptron (MLP) : a foundational neural network architecture.

- Multiple layers of neurons: **input**, **hidden**, and **output.**

Feedforward neural network: all the information flows in one direction.

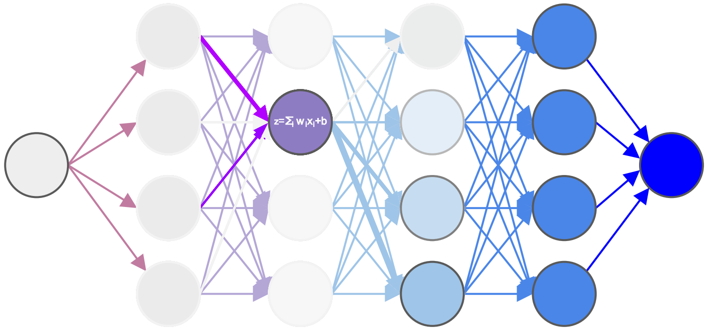


Building block 2:

Activation Functions

Activation functions : nonlinear functions applied to the weighted input sum of each neuron within a layer of a neural network.

- The activation function transforms the weighted input sum into an output signal



1. Receives inputs
2. Multiplies inputs by weights
3. Adds a bias
4. Applies an activation function
5. Produces an output signal

Building block 2:

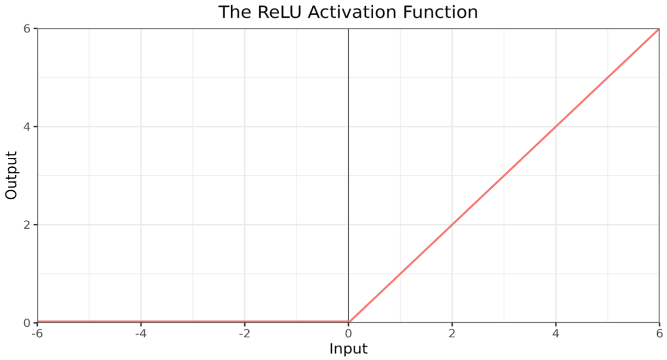
Activation Functions

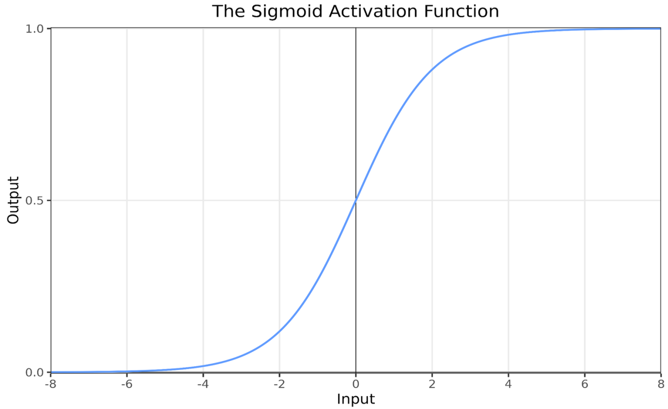
Key activation functions:

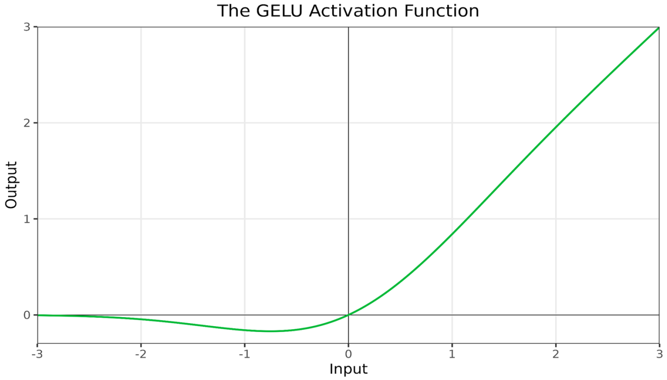
Rectified Linear Unit (ReLU),

Gaussian Error Linear Unit (GELU),

Sigmoid, and Softmax.

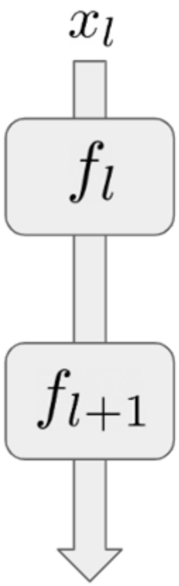






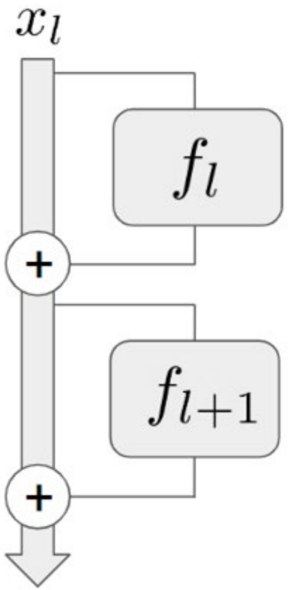
Building block 3:

Residual Connections



Residual connections : pathways for information to bypass network layers.

- Data can skip being transformed in each layer, so its original content can be better preserved.
- Each layer can access more than just the previous layer's representations.
- Lower-level features from earlier layers contribute directly to higher-level features of deeper layers.



By facilitating information flow, they help models learn iterative and hierarchical feature representations.

Building block 4:

Convolution

Convolution : a process used to detect patterns in input data by applying a small matrix called a “filter” or “kernel” and looking for cross-correlations.

Filter

w	x
y	z

Data

aw	bw	cx
dw	ew	fx
gy	hy	iz

→

Activation map

aw + bx + dy + ez	bw + cx + ey + fz
dw + ex + gy + hz	ew + fx + hy + iz

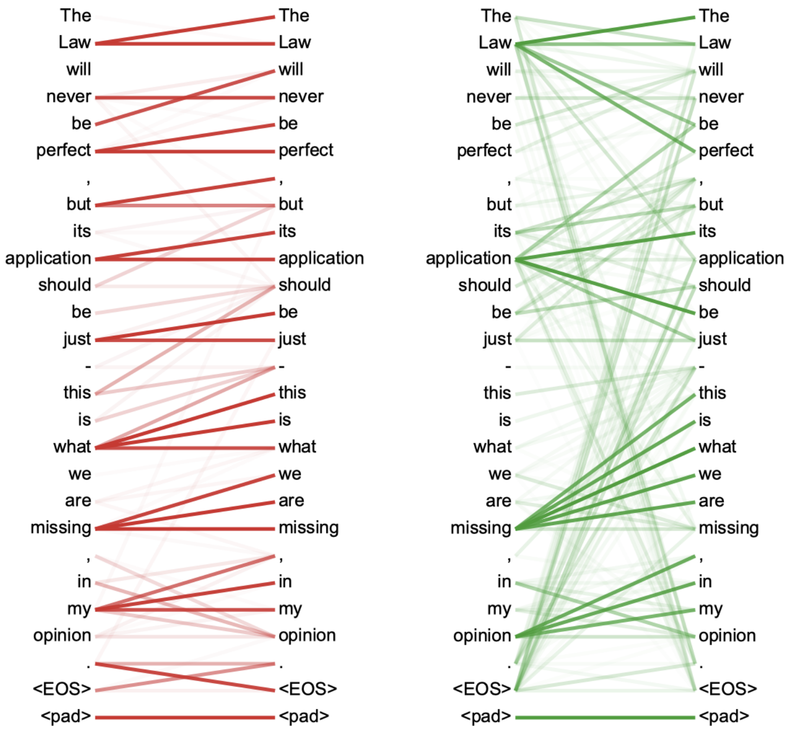
Building block 5:

Self-Attention

Self-attention encodes the relationships between elements in a sequence.

Each element “attends” to every other element by determining its relative importance and focusing on the most relevant connections, capturing relationships within sequences that are separated by long distances.

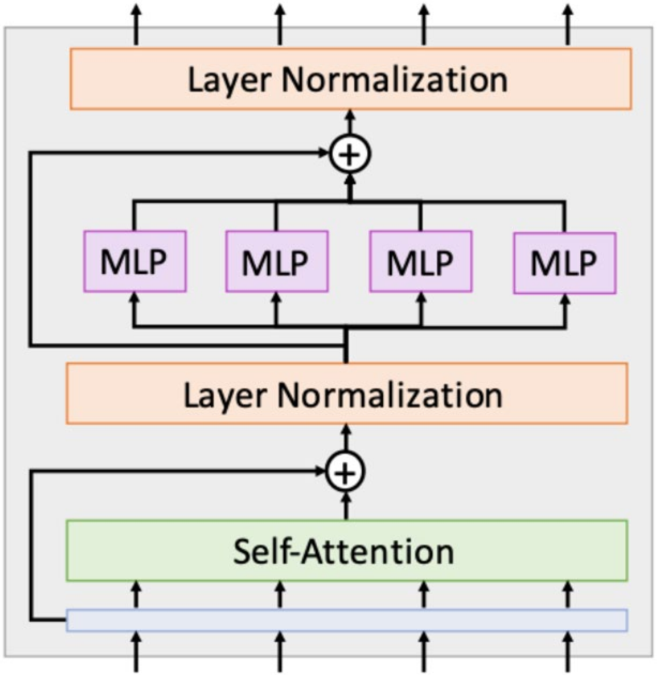
Example: summarizing a long text.



The Transformer

Transformer : a deep learning architecture used in LLMs and other models

The model consists of **Transformer blocks** , which combine **self-attention** and **MLPs** with optimization techniques such as **residual connections** .

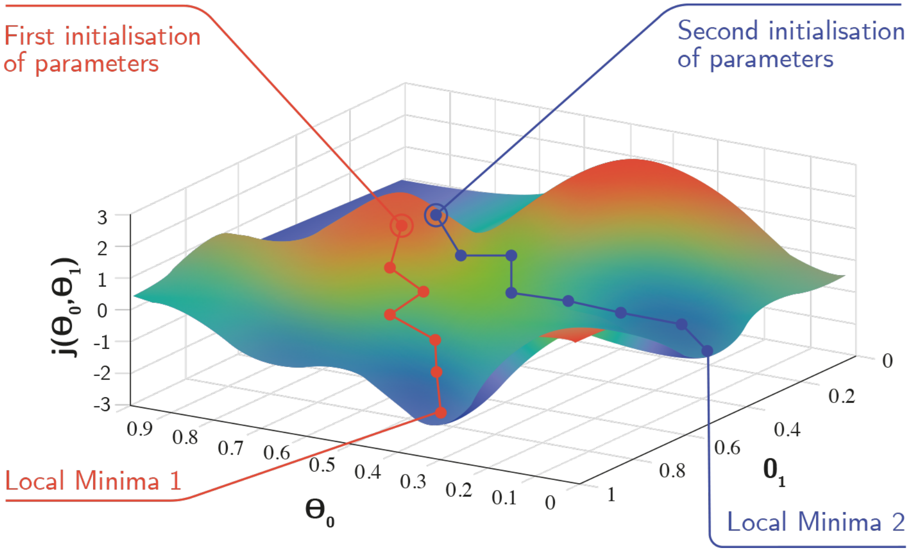


Example: **large language models (LLMs)** :

- Very high parameter numbers (billions)
- Vast quantities of training data
- Most use Transformers to model long-range dependencies effectively.

Training Steps

1. Initialization
2. Forward propagation
3. Loss calculation
4. Backpropagation
5. Parameter update
6. Iteration
7. Stopping criterion



Training Methods (1/2)

Pre-training : training a model on vast quantities of data to give it many generally useful representations.

- These representations are encoded via the model's weights
- Example: training a model to write movie scripts by first teaching it more general rules about grammar and language

Training Methods (2/2)

Fine-tuning : adapting a pre-trained model to a new dataset or task.

- Use the weights of the pre-trained model as a starting point
- Then some (or all) layers are trained on the new task or data, usually backwards from the output layer to the input layer

Few-shot learning : training a model using a small amount of training data.

- Most effective after a model has already learned good representations

Scaling Laws

Many AI systems improve predictably with scale.

Scaling laws

Scaling laws predictably model the how AI systems' performance scales with resources.

Parameter count and dataset size are crucial inputs, both are bottlenecked by compute.

Scaling laws detail the relationship between an AI model's performance and its primary inputs.

Power Laws

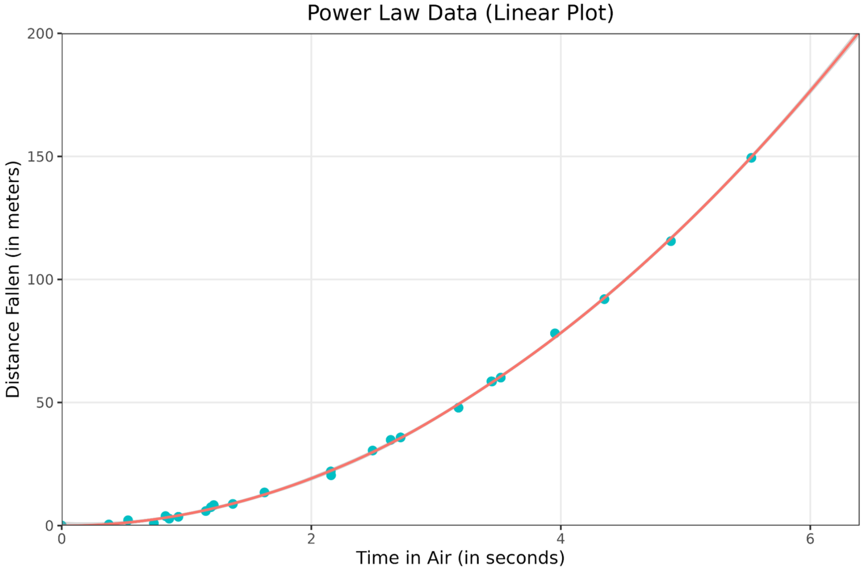
Scaling laws are a type of **power law**, in which one quantity varies as a power of another.

$$y = bx^a$$

The change in y is proportional to the change in x raised to a.

$$\log(y) = a\log(x) + \log(b)$$

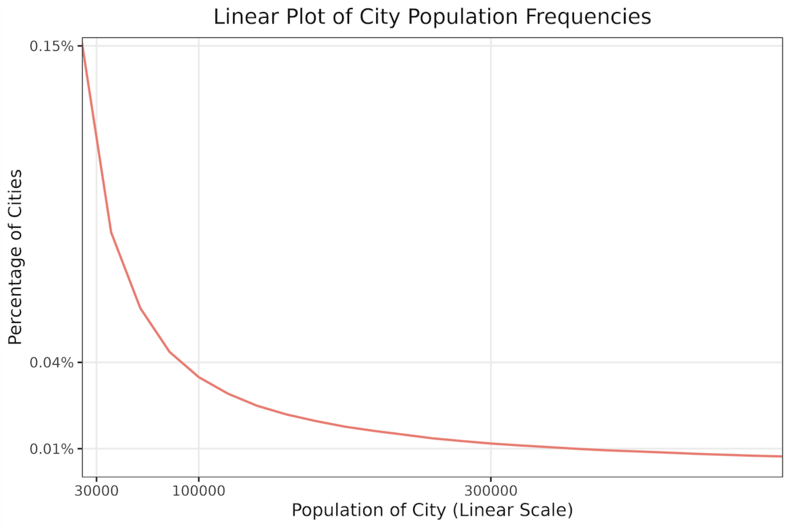
A power law relationship is linear in the logarithmic space.

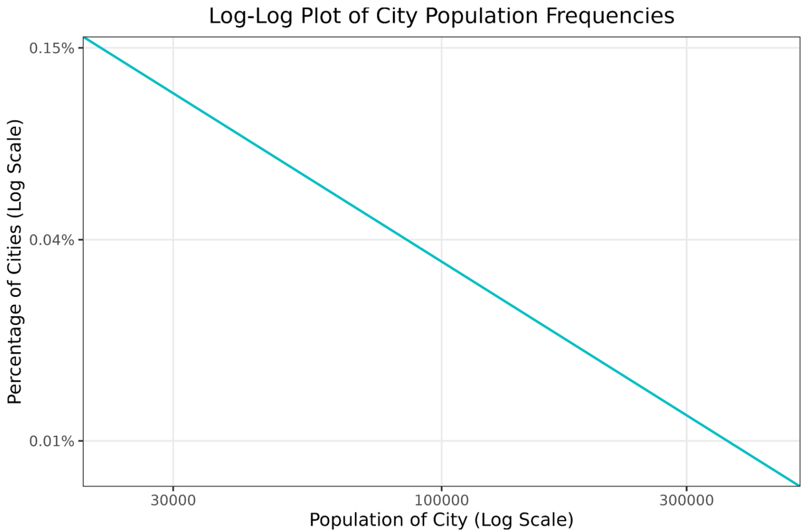


Power Laws

Power laws are very ubiquitous, found throughout nature and society.

Examples include the distributions of earthquake magnitudes, social media content generation, wealth, and city populations.





Scaling Laws in Deep Learning

Scaling laws follow power laws that predict **loss** based on **model size** and **dataset size** .

- Model size is measured in parameters, a model's adjustable weights.
- Dataset size is measured in tokens, fundamental units of data that may vary based on the type of data.

Loss falls nonlinearly with increased numbers of parameters or tokens.

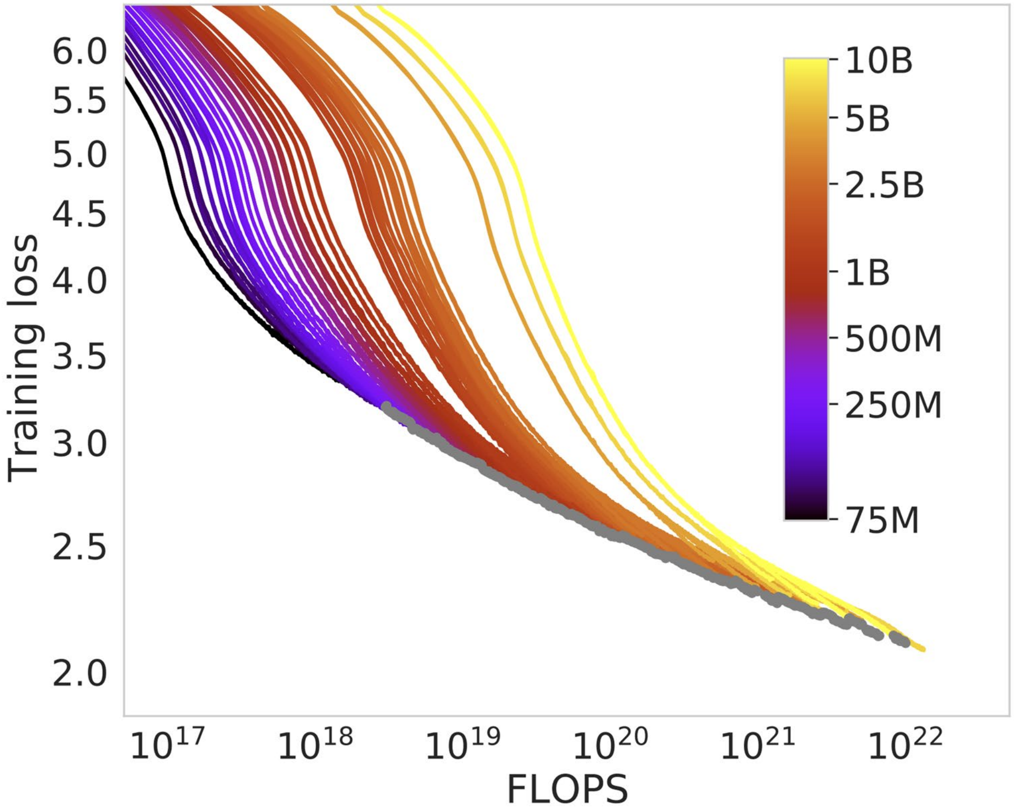
Scaling is dependent on computational power (compute), which enables more operations, more parameters, and processing of more data.

The Chinchilla Scaling Example (1/2)

Chinchilla is a LLM that was outperformed much larger models (in terms of parameters) by training on more data (tokens).

Chinchilla scaling law: $$\underbrace{L(N, D)}_{\text{Loss}} = 406.4\underbrace{N}_{\text{Parameters}}^{-0.34} + 410.7\underbrace{D}_{\text{Tokens}}^{-0.28} + \underbrace{1.69}_{\text{Irreducible Error}}$$

The Chinchilla Scaling Example (2/2)



Discussion of Scaling Laws

Scaling laws hold across modalities and orders of magnitude—factors of ten—applying to both small and larger models.

Scaling laws describe some ML models more closely than others.

- Generative models like LLMs following scaling laws closely.
- Discriminative models like image classifiers do not.

Better learning algorithms increase the constant term in scaling laws.

The Bitter Lesson

AI researchers have often tried to improve systems using expert knowledge.

This approach only leads to improvements in the short term, and may even hurt long term progress.

Breakthrough progress occurs when developers instead scale computation and data.

Takeaway: The simple method of increasing scale beats complex research approaches.

Chapter Recap

Artificial Intelligence: Definitions, historical development, and types.

Machine Learning: Models, tasks, data, and learning types.

Deep Learning: Fundamental concepts and applications.

Scaling Laws: Predictable improvement in performance with scale.