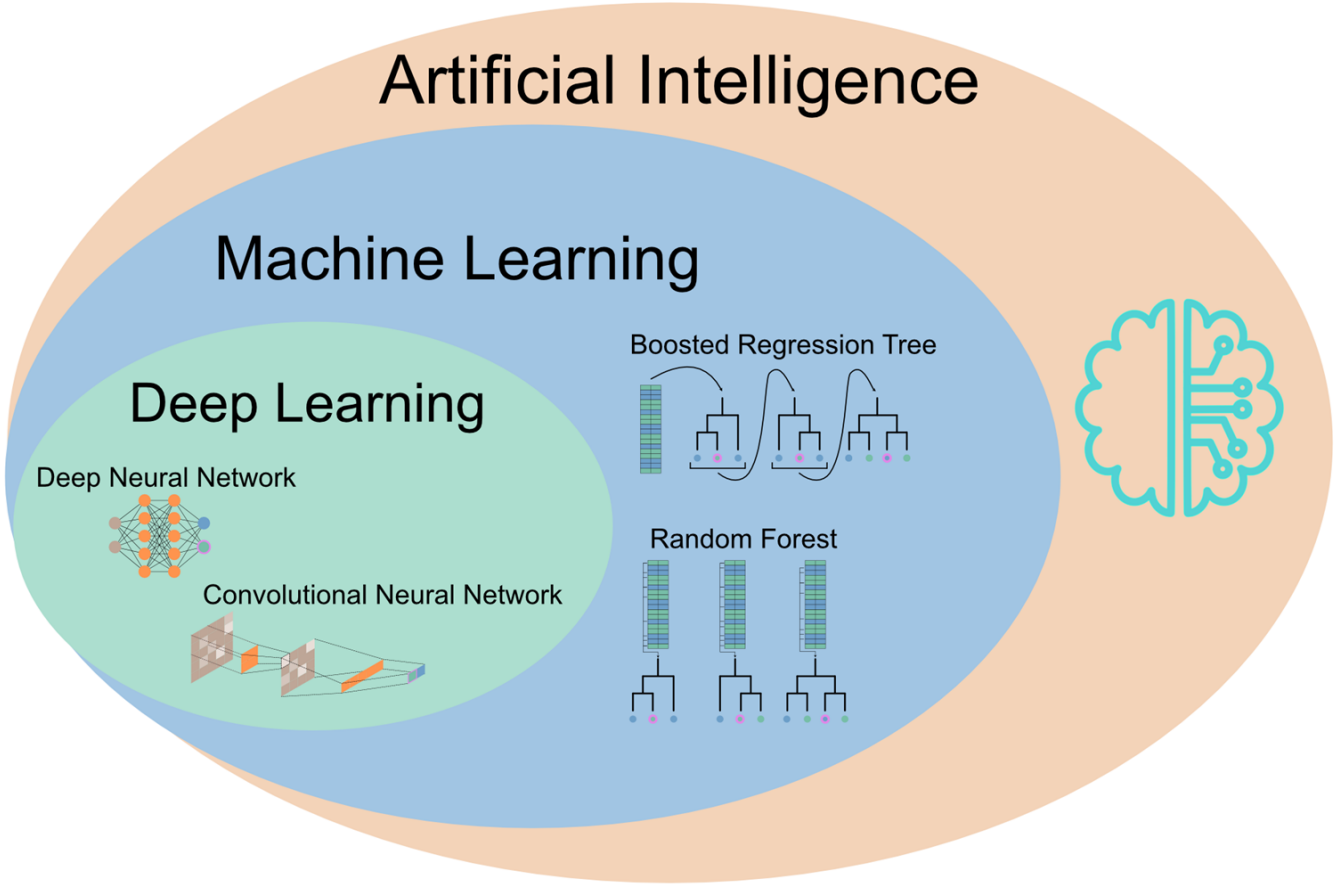
BIG DATA ANALYTICS (CS-431)

Dr. Sriparna Saha
Associate Professor

Website: https://www.iitp.ac.in/~sriparna/
Google Scholar: https://scholar.google.co.in/citations?user=Fj7jA_AAAAAJ&hl=en
Research Lab: SS_Lab
Core Research AREA: NLP, GenAI, LLMs, VLMs, Multimodality, Meta-Learning, Health Care, FinTech, Conversational Agents

TAs: Sarmistha Das, Nitish Kumar, Divyanshu Singh, Aditya Bhagat, Harsh Raj


Artificial Intelligence
Machine Learning
Deep Learning
Deep Neural Network
Convolutional Neural Network
Boosted Regression Tree
Random Forest

1. Advanced Machine Learning Techniques

Big Data is high-volume, high-velocity, and high-variety, so advanced ML approaches focus on scalability and efficiency.

- **Ensemble Learning** (Random Forests, Gradient Boosting, XGBoost, LightGBM): Effective for high-dimensional structured data).
- **Online & Incremental Learning** (e.g., Hoeffding Trees, Online SVMs): Suitable for real-time data streams.
- **Distributed ML** (Apache Spark MLlib, FlinkML): Enables training ML models across large-scale clusters.
- **Feature Engineering at Scale** (AutoML, Feature Stores like Feast): Automates feature selection/creation for Big Data pipelines.
- **Semi-supervised & Weakly Supervised Learning**: Leverages partially labeled or noisy data, common in big datasets.

Ensemble Learning

Ensemble learning combines multiple learners to improve predictive performance. It has been adopted in response to issues resulting from limited datasets.

Why use ensemble learning?

Bias-variance tradeoff

- Bias measures the average difference between predicted values and true values. As bias increases, a model predicts less accurately on a training dataset. High bias refers to high error in training. Optimization signifies attempts to reduce bias.

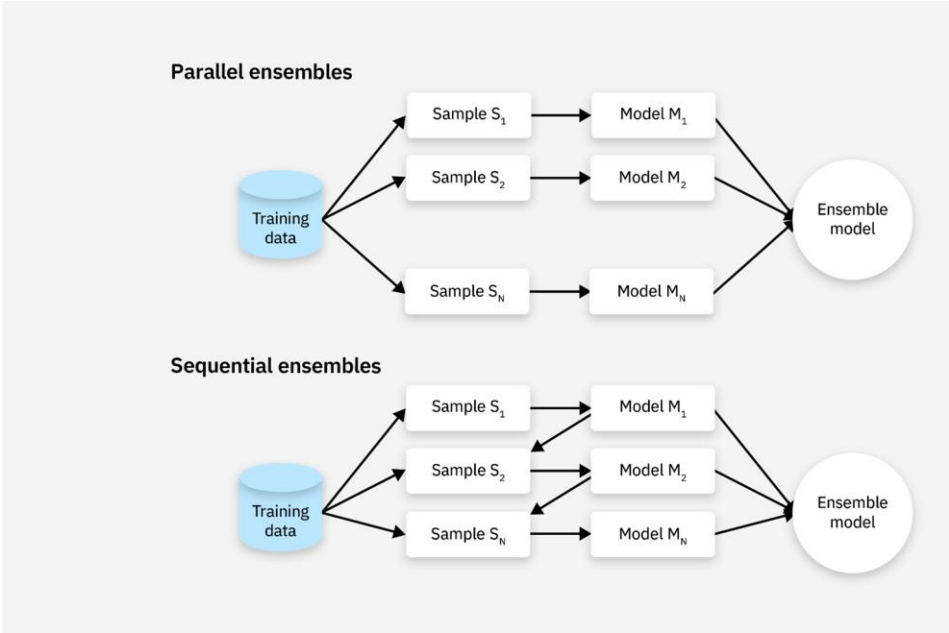
- Variance measures the difference between predictions across various realizations of a given model. As variance increases, a model predicts less accurately on unseen data. High variance refers to high error during testing and validation. Generalization refers to attempts to reduce variance.

$$Error = Bias^2 + Variance + Irreducible\ Error$$

Types of Ensemble Learning

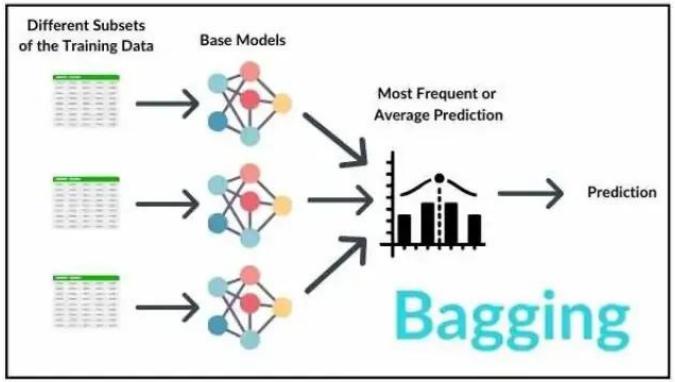
- **Parallel** methods train each base learner apart from the others of the others. Per its name, then, parallel ensembles train base learners in parallel and independent of one another.

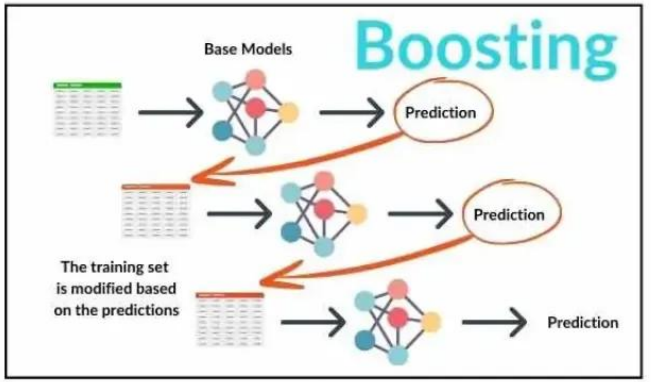
- **Sequential** methods train a new base learner so that it minimizes errors made by the previous model trained in the preceding step. In other words, sequential methods construct base models sequentially in stages.[9]

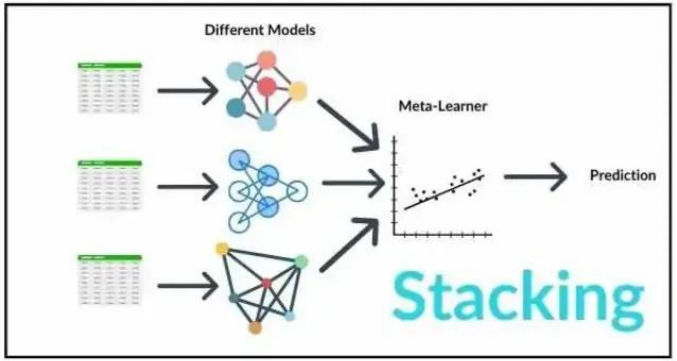


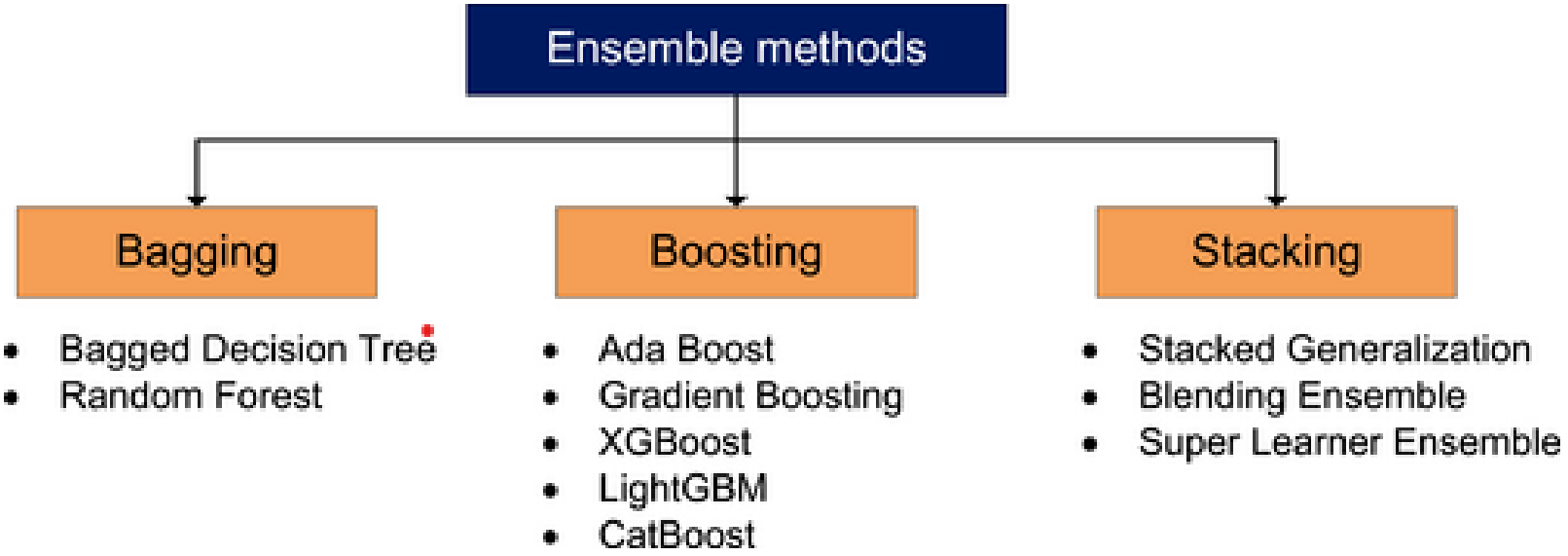
How do ensemble m

Some techniques like stacking train a meta-learner on base models, but the most common approach is majority voting.








Ensemble methods
Bagging
Boosting
Stacking
Bagged Decision Tree
Random Forest
Ada Boost
Gradient Boosting
XGBoost
LightGBM
CatBoost
Stacked Generalization
Blending Ensemble
Super Learner Ensemble

Random Forest Regression (Bagging)

- **Base learner used in the Super Learner stack**
- Idea: average predictions from many bootstrapped decision trees to reduce variance.
- Why here: captures nonlinear mix–strength relations while resisting overfitting.
- Key knobs: n_estimators, max_depth, min_samples_split, max_features.
- Training: grid-searched hyperparameters; strong baseline among ensembles.
- Pros/Cons: robust & interpretable via feature importance; larger models can be slower.

From Lee et al. (2023), used as one of six base ensembles before stacking.

AdaBoost Regression

- **Boosting with reweighting of residuals**
- Idea: sequentially reweight errors; new weak learners focus on hard-to-predict mixes.
- Weak learner: shallow trees (stumps) commonly used for stability.
- Key knobs: n_estimators, learning_rate; sensitive to noise/outliers.
- Use-case fit: can learn monotone trends (e.g., w/b vs. strength) but may underfit strong nonlinearity alone.
- Role in study: complementary signal feeding the meta-learner.

From Lee et al. (2023); part of the stacked Super Learner.

Gradient Boosting Machine (GBM)

- **Stage-wise residual fitting with shrinkage**
- Idea: stage-wise additive trees fit to negative gradients (residuals).
- Controls bias–variance via learning_rate, n_estimators, max_depth (shallow trees).
- Captures complex interactions between constituents (cement, SCMs, w/b, curing age).
- Typically stronger than AdaBoost on tabular regression; risk of overfitting without shrinkage & early stopping.
- Provides calibrated residual patterns for stacking.

From Lee et al. (2023); tuned via grid search before stacking.

Extreme Gradient Boosting (XGBoost)

- **Regularized, high-performance GBM**
- Enhancements: regularized objective (L1/L2), column/row subsampling, efficient tree growth.
- Often the strongest single model on structured data with careful tuning.
- Key knobs: learning_rate, max_depth, subsample, colsample_bytree, n_estimators, reg_alpha/lambda.
- In HPC data, balances bias & variance and handles feature collinearity.
- Strong base learner signal for the Super Learner meta-model.

From Lee et al. (2023); one of the top-performing base learners.

LightGBM

- **Histogram-based, leaf-wise boosting**
- Leaf-wise tree growth with depth constraints for speed & accuracy on large/tabular data.
- Histogram-based splits accelerate training; good with continuous mix features.
- Key knobs: num_leaves, max_depth, learning_rate, feature_fraction, bagging_fraction.
- Tends to excel when interactions are sparse but strong.
- Provides diverse error patterns relative to XGBoost → valuable in stacking.

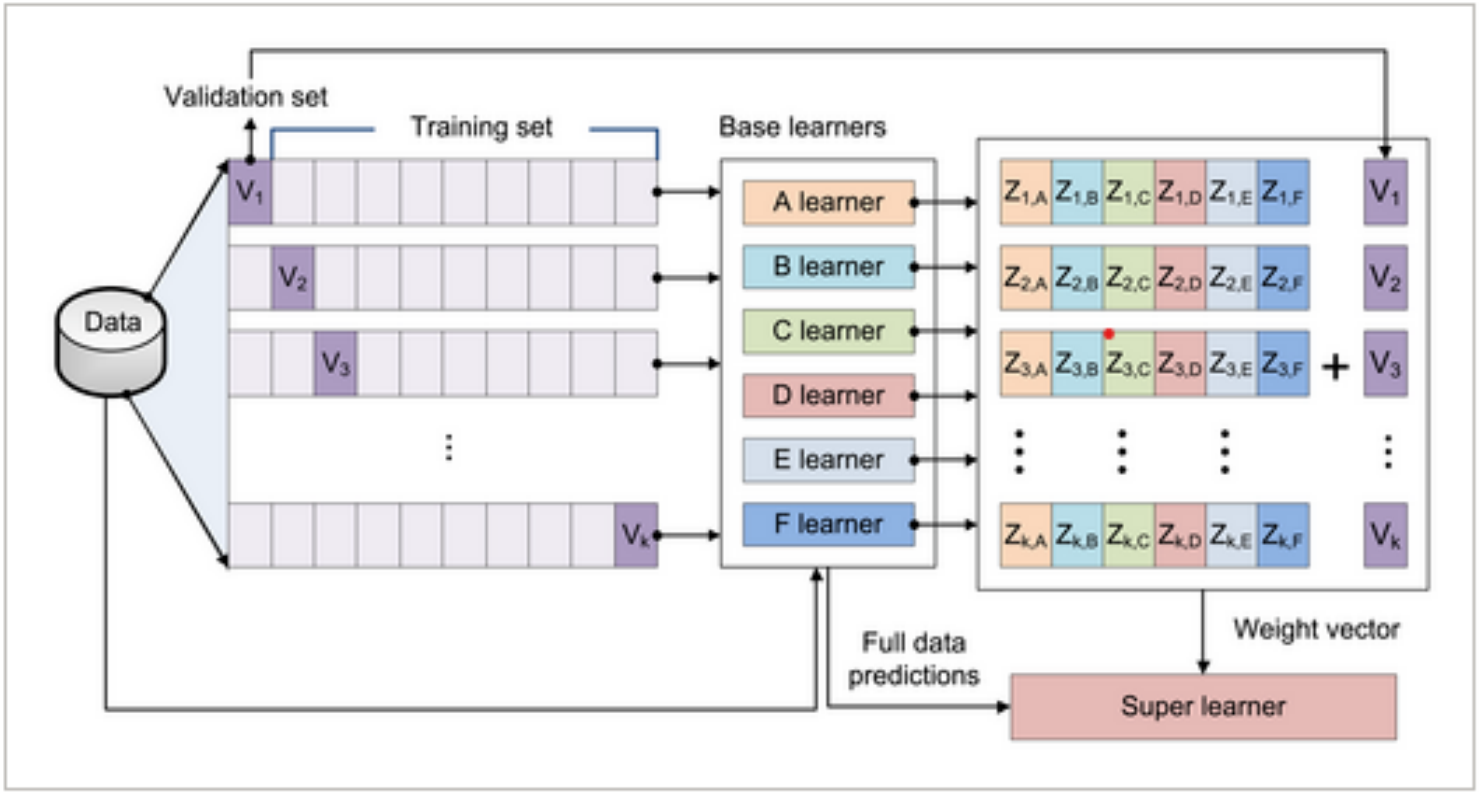
From Lee et al. (2023); tuned and used in the stacked ensemble.

CatBoost

- **Ordered boosting with symmetric trees**
- Order-boosting + symmetric trees; robust training dynamics and reduced prediction shift.
- Natively handles categorical variables; still competitive on fully numeric data.
- Key knobs: depth, learning_rate, l2_leaf_reg, iterations; uses ordered boosting.
- In concrete datasets (mostly numeric), adds diversity in residuals for stacking.
- Complements tree-based ensembles within the Super Learner.

From Lee et al. (2023); contributes complementary patterns for the meta-learner.

The super learner (SL) ensemble method



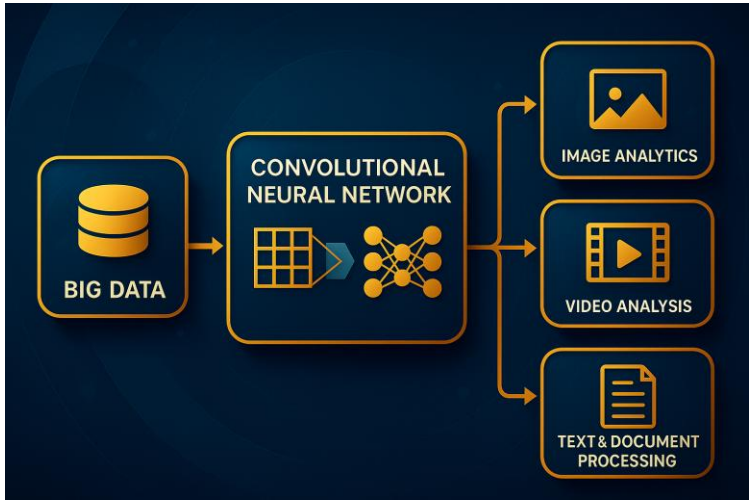
2. Deep Learning Techniques

Deep learning is crucial when working with **unstructured Big Data** (text, images, video, audio).

- **CNNs (Convolutional Neural Networks):** For large-scale image/video analytics.
- **RNNs / LSTMs / GRUs:** For time-series, sequential logs, and IoT data streams.
- **Transformers (BERT, GPT, Vision Transformers):** For large-scale NLP and multimodal big data.
- **Autoencoders & Variational Autoencoders (VAEs):** Dimensionality reduction and anomaly detection in large datasets.
- **Graph Neural Networks (GNNs):** For social networks, fraud detection, and recommendation in massive graph data.
- **Reinforcement Learning (RL):** Applied in large-scale optimization problems (e.g., traffic systems, financial trading).

Why CNNs in Bigata Analytics?

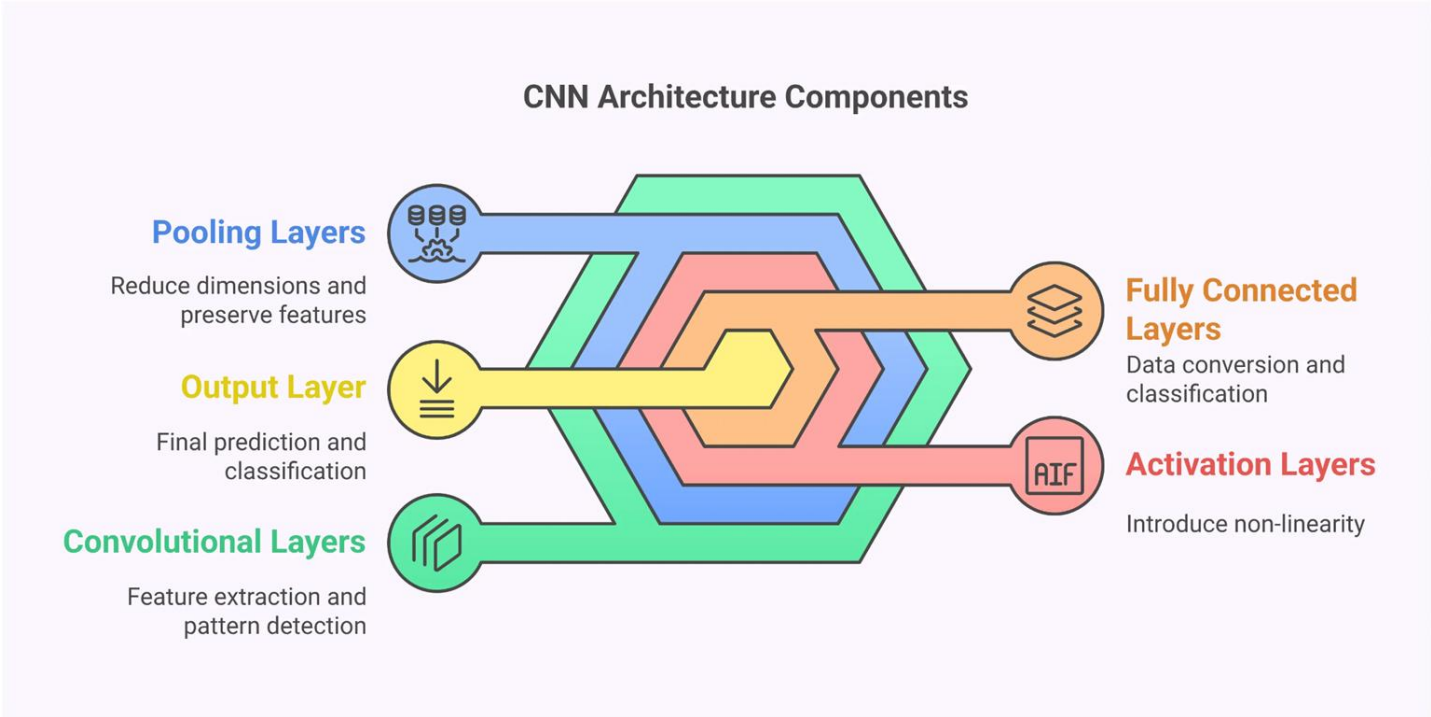
Big Data isn't just numbers—it includes **images, videos, audio, and text**. CNNs are designed to automatically extract hierarchical features (edges → shapes → objects → context) from such **high-dimensional unstructured data**, which traditional ML struggles with.



Applications of CNNs in Big Data

- **Image Analytics:** Medical imaging (MRI/CT scans), satellite imagery (climate, agriculture), facial recognition at population scale.
- **Video Analysis:** Large-scale surveillance, fraud detection in finance (analyzing video transactions), entertainment platforms (e.g., YouTube recommendation).
- **Text & Document Processing (via CNNs for NLP):** Large document corpora, sentiment mining, customer reviews.
- **Sensor/IoT Data:** CNNs can analyze spatio-temporal patterns in traffic, smart grids, or industry sensors.

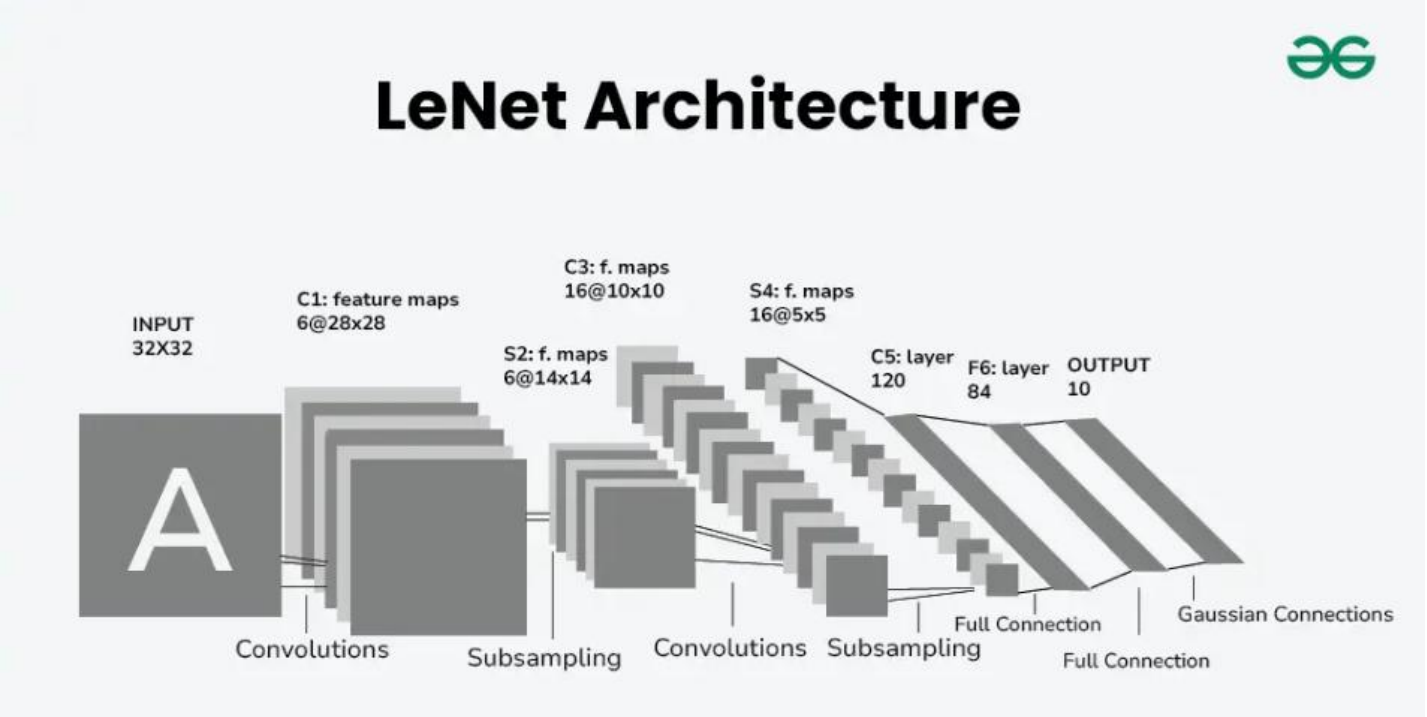
Convolutional Neural Networks



Types of CNNs

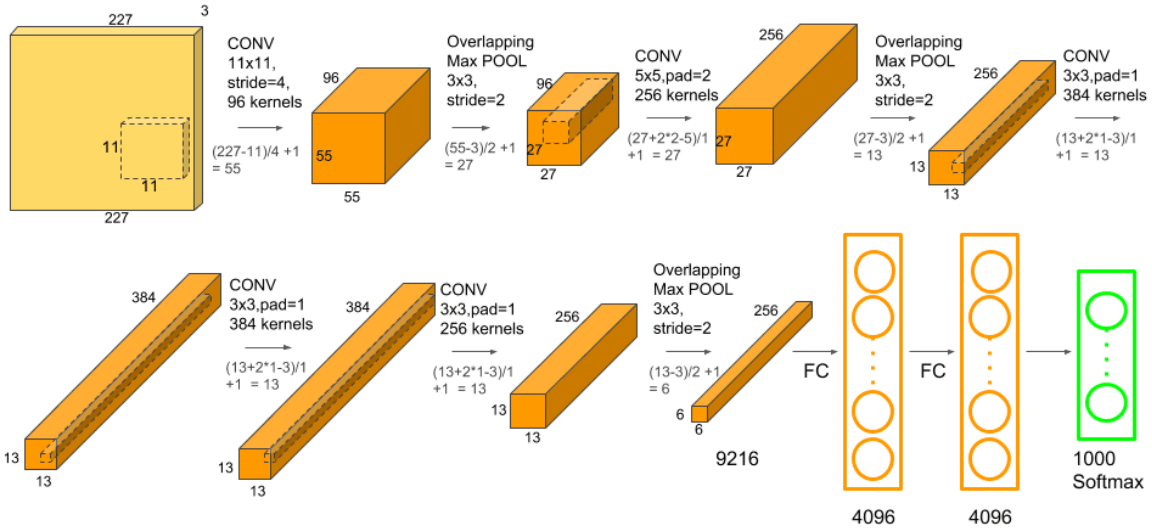
Use case: Early-stage image classification.

Big Data Relevance: Foundation for CNN architectures; useful in benchmarking and preprocessing before scaling to big data systems.



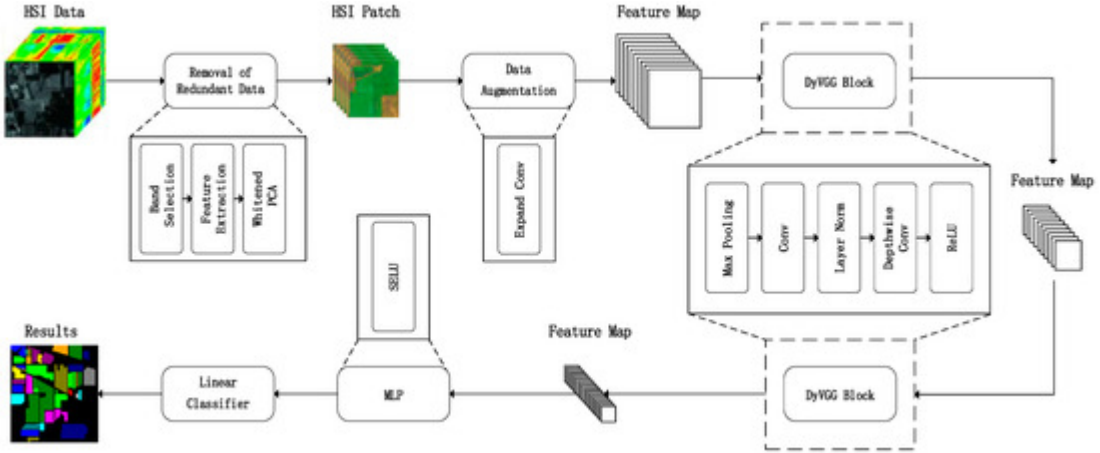
AlexNet

- **Use case**: Large-scale image classification (ImageNet).
- **Big Data Relevance**: Introduced ReLU activation and GPU training, making CNNs practical for massive datasets like ImageNet (millions of images).



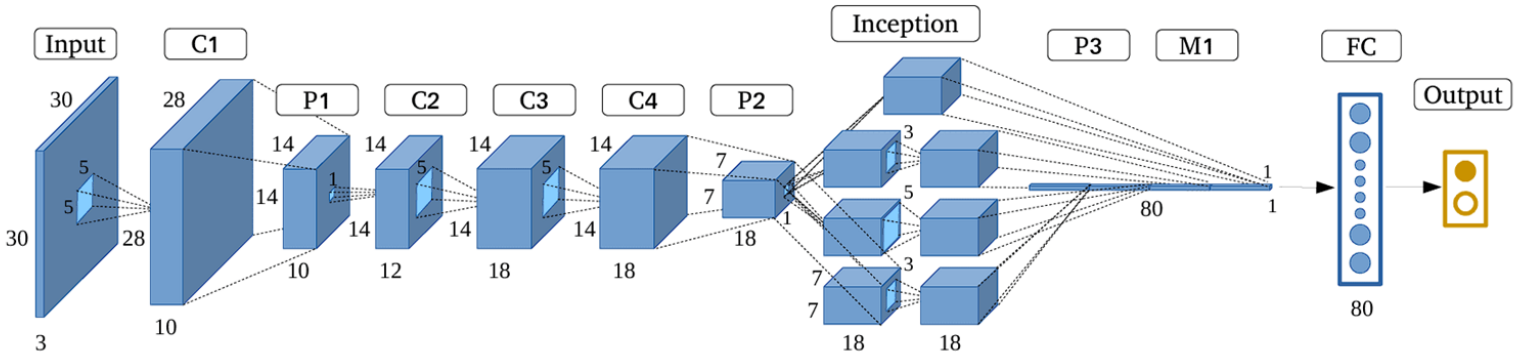
VGGNet

- **Use case**: Deep hierarchical feature extraction.
- **Big Data Relevance**: Uses many convolutional layers with small (3×3) filters, effective for detailed feature learning in high-dimensional big datasets.



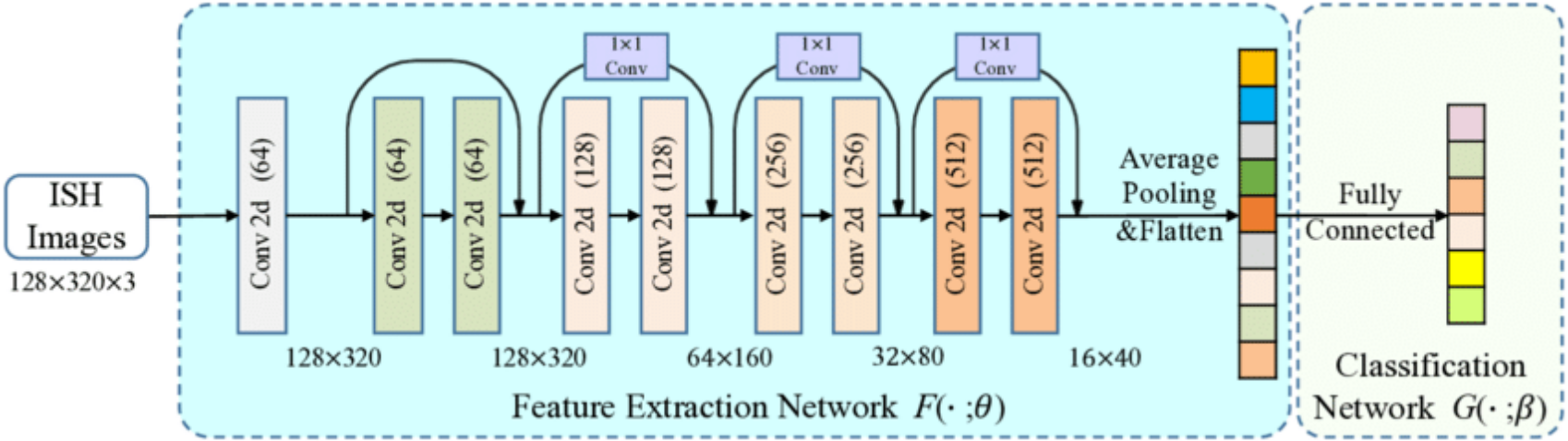
GoogLeNet (Inception Network)

- **Use case**: Efficient large-scale image/video analytics.
- **Big Data Relevance**: Inception modules reduce computational cost while handling large-scale multimedia datasets efficiently



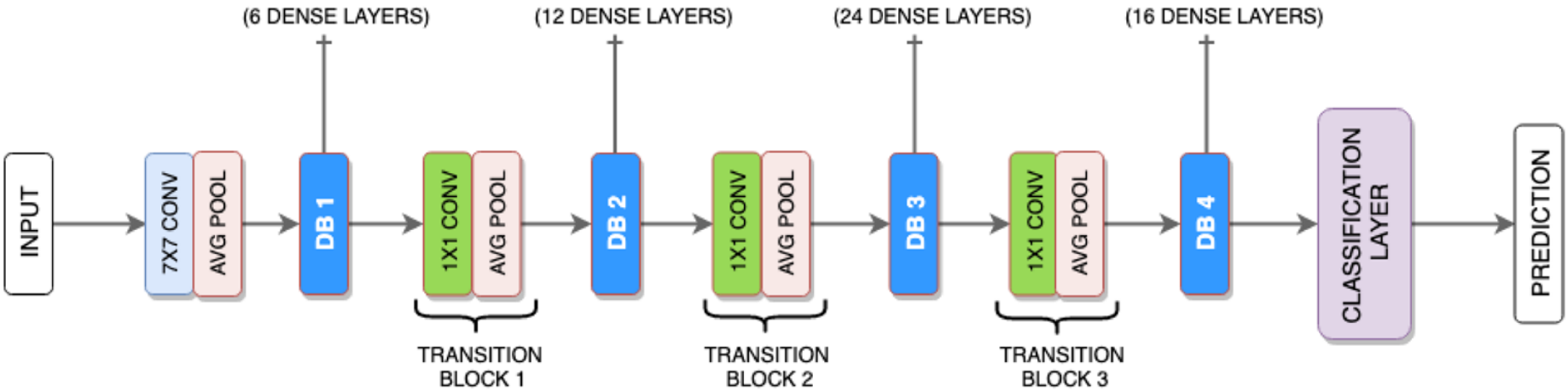
ResNet (Residual Networks)

- **Use case**: Very deep networks (50–152 layers).
- **Big Data Relevance**: Skip connections solve vanishing gradient problems, making it scalable for massive, high-dimensional data in big data systems.



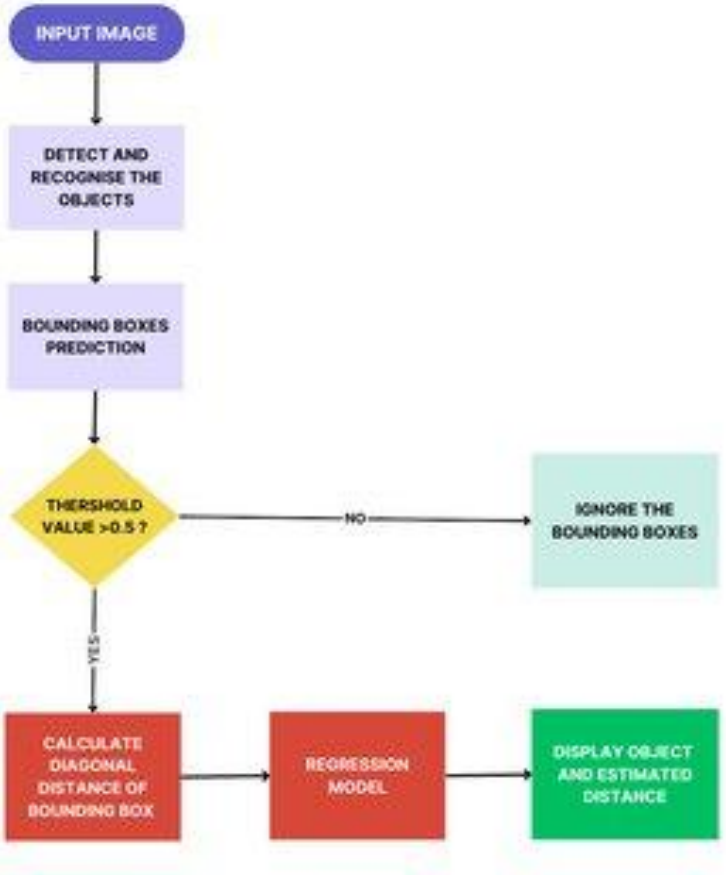
DenseNet

- **Use case**: Efficient feature reuse.
- **Big Data Relevance**: Each layer is connected to every other layer, reducing parameters while handling complex multimodal big data analytics.



MobileNet / EfficientNet

- **Use case**: Lightweight, scalable CNNs.
- **Big Data Relevance**: Optimized for real-time big data analytics on edge devices (IoT, mobile, healthcare).

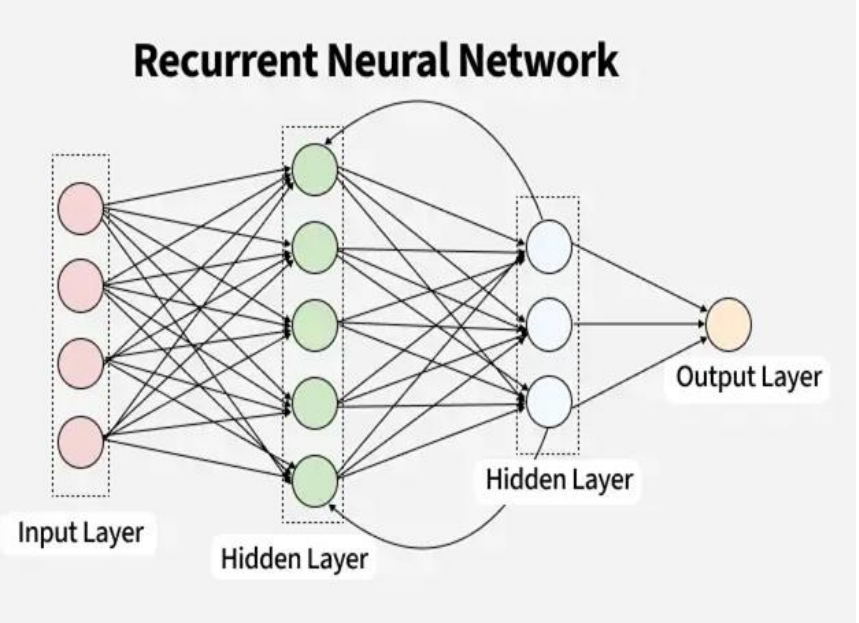


Recurrent Neural Networks (RNNs)

Understanding Sequential Big Data

Much of big data is not static; it's a sequence. RNNs were designed to process this data by maintaining an internal "memory" of past information.

- **Core Idea:** An RNN processes a sequence step-by-step, using a feedback loop to pass information from the previous step to the current one. This gives it a form of memory.
- **Big Data Applications:**
  - **Time-Series Analysis:** Predictive maintenance on IoT sensor data from machinery.
  - **Transaction Analysis:** Detecting fraudulent patterns in a sequence of financial transactions.
  - **User Behavior:** Predicting user churn based on their clickstream activity over time.

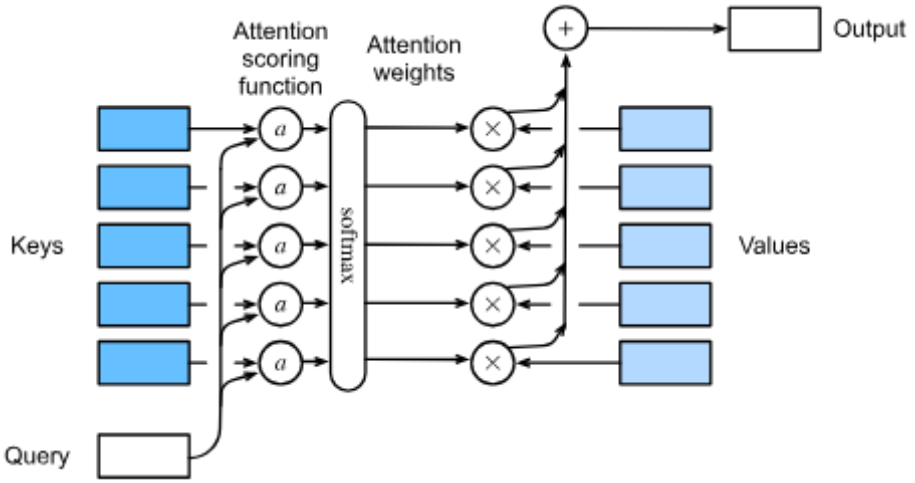


The Attention Mechanism

A Breakthrough: Focusing on What's Important

A key limitation of simple RNNs is their difficulty in remembering information over long sequences. The **Attention Mechanism** solves this by allowing the model to dynamically focus on the most relevant parts of the input data.

- **Core Idea:** Instead of relying on a single summary of the past, the model learns to assign "attention scores" to all previous inputs. It gives higher scores (more focus) to the inputs that are most relevant for the current prediction.
- **Big Data Relevance:** When summarizing a massive document, the model can attend to the key sentences, ignoring filler text. In time-series forecasting, it can focus on past events that were most influential.

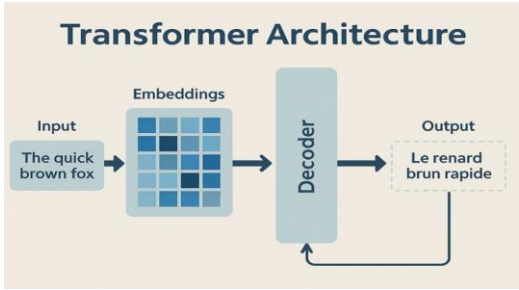


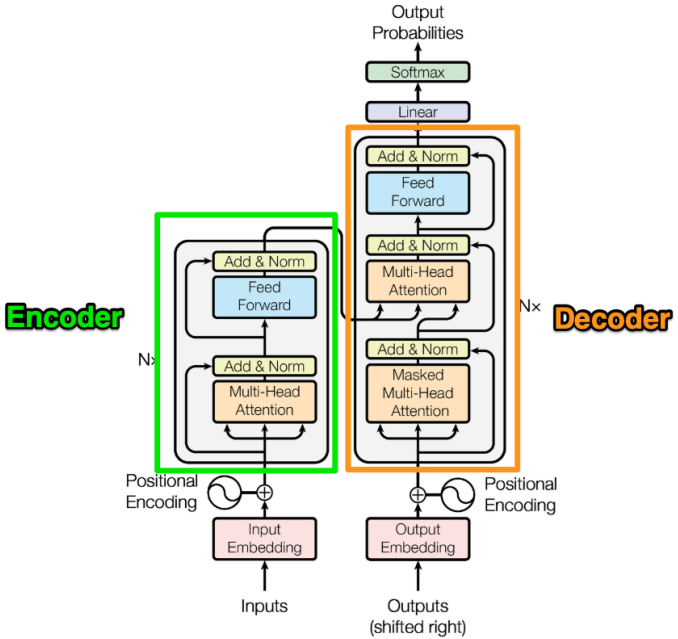
Transformers: The Modern Architecture

Rethinking Sequences with Self-Attention

The Transformer architecture, introduced in "Attention Is All You Need," revolutionized sequence modeling by removing recurrence and relying entirely on attention.

- **Self-Attention:** Allows the model to weigh the importance of all other words in the *same* sequence. This helps it understand context, grammar, and complex relationships (e.g., linking a pronoun to the noun it represents).
- **Multi-Head Attention:** Runs the self-attention process multiple times in parallel, allowing the model to learn different types of relationships simultaneously from various perspectives.
- **Key Advantage for Big Data:** Transformers are highly **parallelizable**. Unlike RNNs, they can process an entire sequence at once, drastically reducing training time on massive datasets.



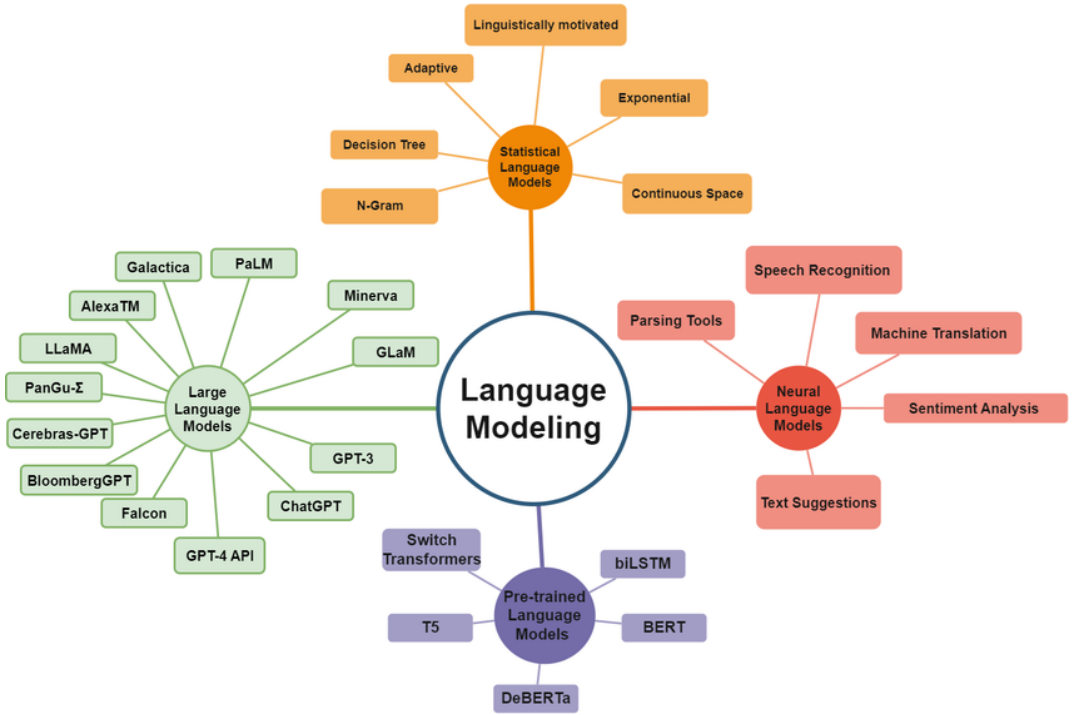


Language Models (LMs)

Applying Transformers to Unstructured Data

Language Models are a direct and powerful application of the Transformer architecture. By training on vast amounts of text data, they become versatile tools for a wide range of analytics tasks.

- **Core Task:** Predicting the next word in a sequence. By doing this over a massive corpus (like the internet), they learn grammar, facts, reasoning, and context.
- **Role in Big Data:** They serve as the foundational engine for understanding and interacting with the enormous volumes of unstructured text data found in any large organization—from emails and reports to customer reviews and support tickets.

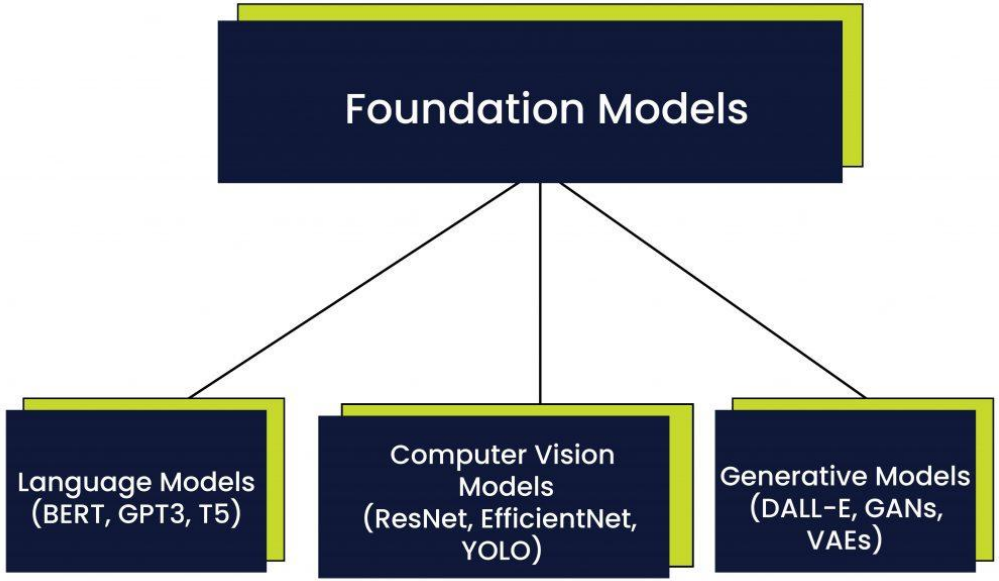


Foundation Models

Large, Pre-Trained Models for General Tasks

Foundation Models (like GPT, LLaMA, Claude, Gemini) are massive models trained on web-scale data. They are not built for one specific task but serve as a powerful base that can be adapted for many purposes.

- **Big Data Applications:**
  - **Large-Scale Summarization:** Condensing thousands of financial reports or legal documents into concise, actionable summaries.
  - **Semantic Search & Q&A:** Building systems that can answer complex questions by understanding the content within an entire corporate knowledge base.
  - **Code Generation:** Assisting data analysts by generating SQL queries or Python scripts from natural language prompts.

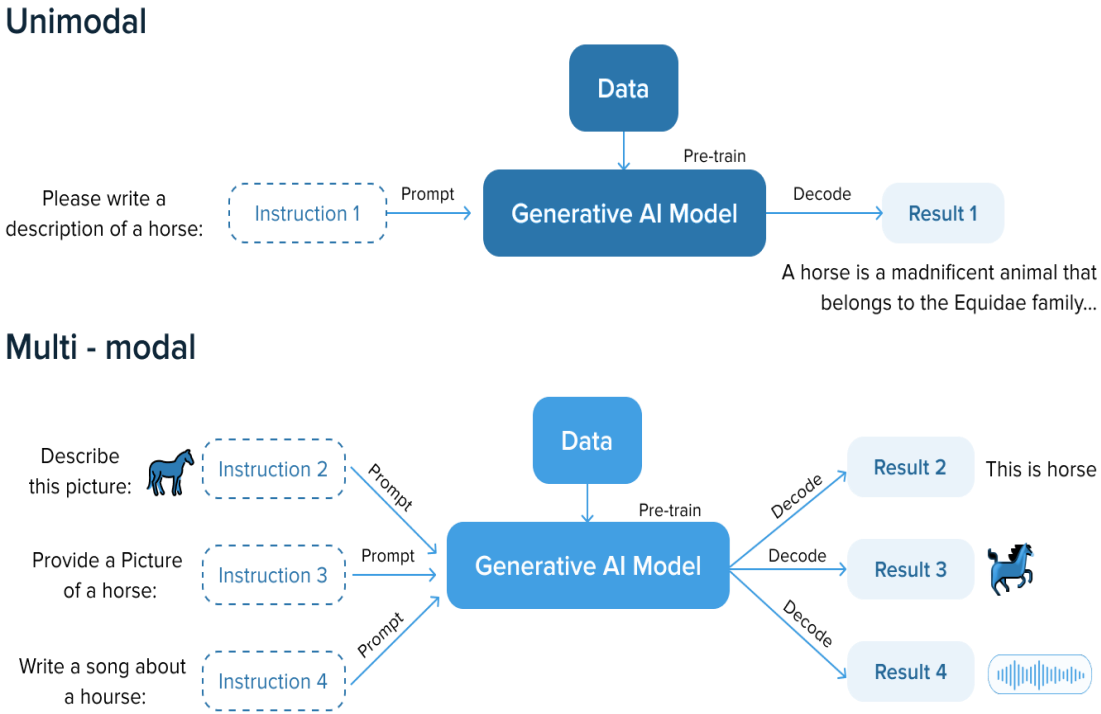


Multimodal Gen-AI

Beyond Text: A Unified View of Data

Big data is inherently multimodal—it's a mix of text, images, videos, and audio. Multimodal Gen-AI models (like CLIP, BLIP) are designed to understand the relationships between these different data types.

- **Core Capability:** They learn a shared "representation space" where, for example, the image of a dog and the text "a photo of a dog" are located close together.
- **Big Data Applications:**
  - **Unified Data Lake Analysis:** Searching across a data lake containing images, videos, and documents with a single text query.
  - **Enhanced Customer Analytics:** Combining product images with text-based customer reviews to gain deeper insights into sentiment and quality issues.

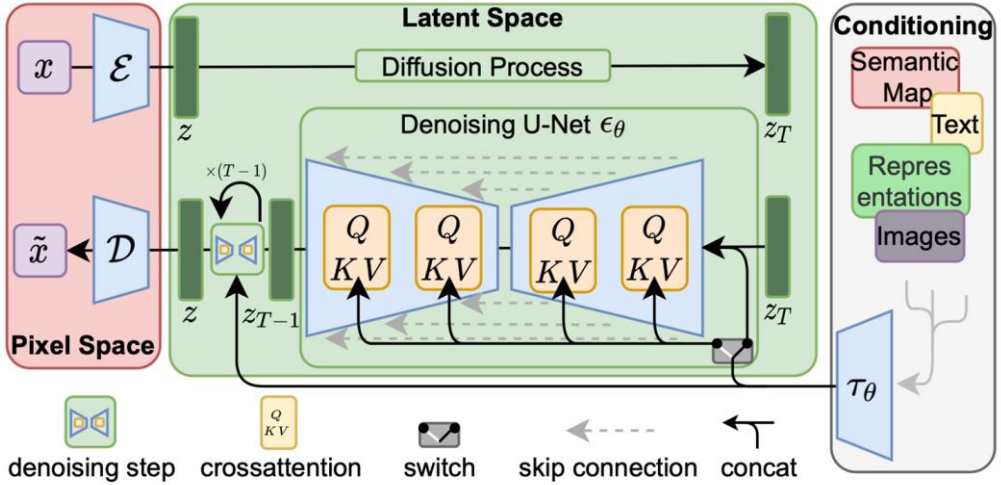


Diffusion Models

Generating High-Fidelity Visual Data

Diffusion Models (like Stable Diffusion, Imagen) are a class of generative models that excel at creating realistic images from text prompts. They work by starting with random noise and gradually refining it into a coherent image.

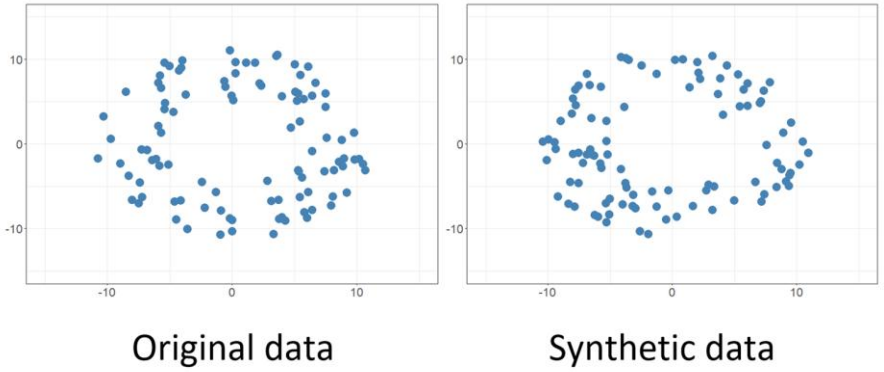
- **Primary Big Data Application: Synthetic Data Augmentation**
  - In many real-world datasets, critical events are rare (e.g., a specific manufacturing defect, a rare medical condition in an X-ray).
  - Diffusion models can generate thousands of photorealistic examples of these rare cases, creating a more balanced and robust dataset for training computer vision models.

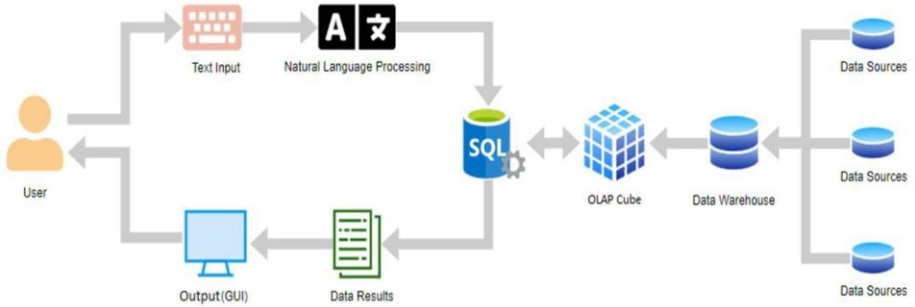


Synthetic Data & Natural Language Querying

Improving Privacy and Accessibility

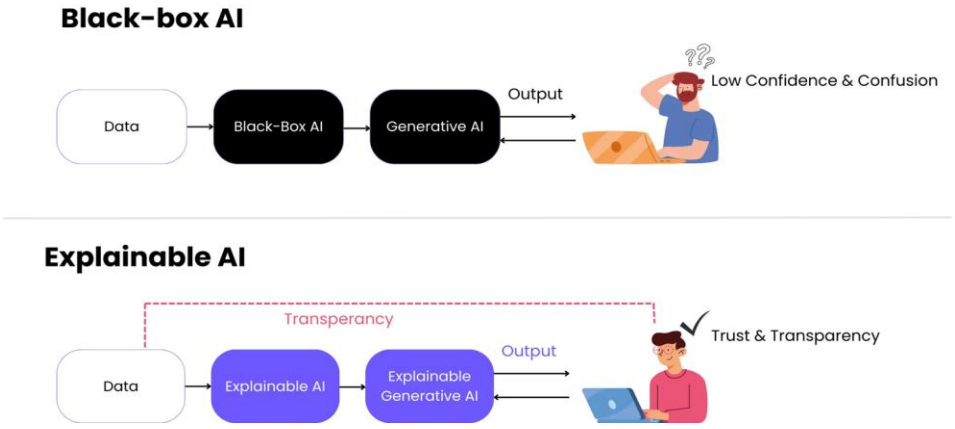
- **Synthetic Data Generation:**
  - **Purpose:** Generates realistic but artificial **tabular datasets** that mimic the statistical properties of real data.
  - **Big Data Relevance:** This is critical in regulated industries like **healthcare and finance**. It allows data scientists to develop and test models without accessing sensitive, personally identifiable information (PII), thus preserving privacy.
- **Gen-AI for Querying Big Data:**
  - **Purpose:** Provides a natural language layer on top of complex databases.
  - **Big Data Relevance:** Tools like **Text-to-SQL** translate plain English questions ("What were our top 10 products in Europe last quarter?") into executable SQL code. This democratizes data access for





Explainable Gen-AI Building Trust in Generative Systems

- As generative models are used for more critical decisions, the "black box" nature becomes a significant risk. Greater transparency is needed to make these systems more trustworthy.
- **The Problem:** Why did the model summarize a document this way? Why did it generate this specific data point? Without answers, it's hard to trust the output.
- **Big Data Relevance:** Transparency is essential for debugging, ensuring fairness, and meeting regulatory compliance. It provides methods to trace a model's output back to the input data, helping analysts understand and validate the results for critical big data decisions.

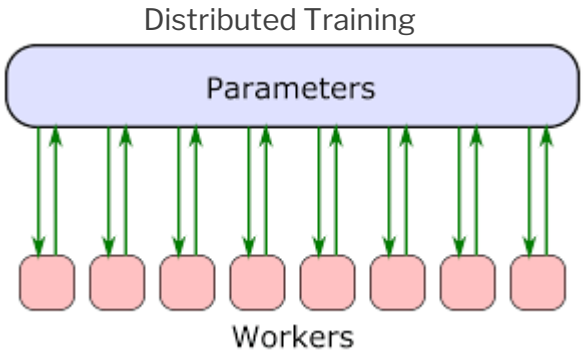


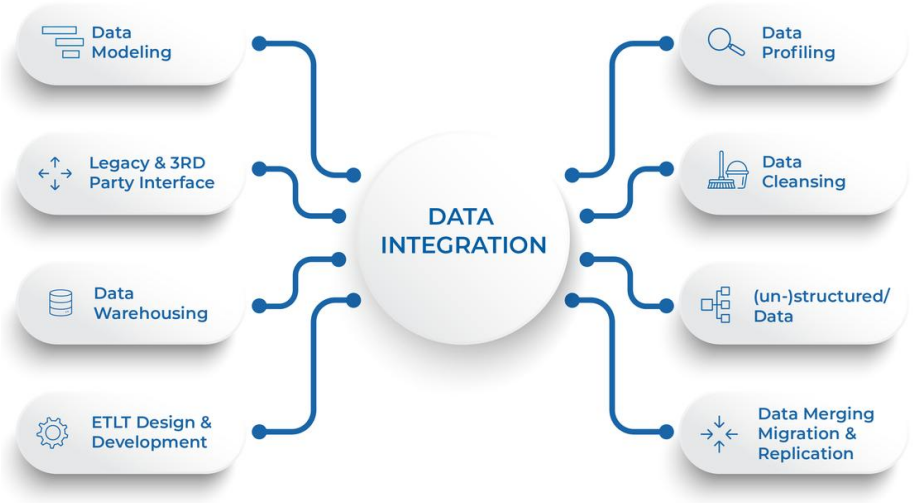
Distributed Training & Data Integration

Scaling Up the Foundation

Advanced models require massive compute power and seamless access to data.

- **Distributed Training Frameworks (TensorFlowOnSpark, Horovod, PyTorch DDP):**
  - **Purpose:** When a model or dataset is too large for one machine, these frameworks orchestrate the training process across a cluster of multiple computers, handling data parallelism and model synchronization.
- **Data Lake & Warehouse Integration (Snowflake, Delta Lake, BigQuery ML):**
  - **Purpose:** The modern paradigm is to **bring compute to the data**. These platforms allow you to train and deploy ML models directly within the data warehouse, minimizing costly and insecure data movement.

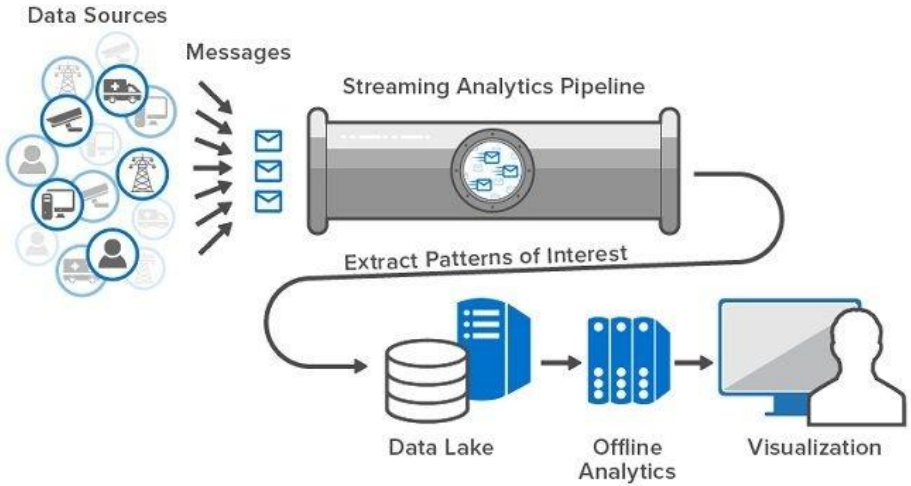


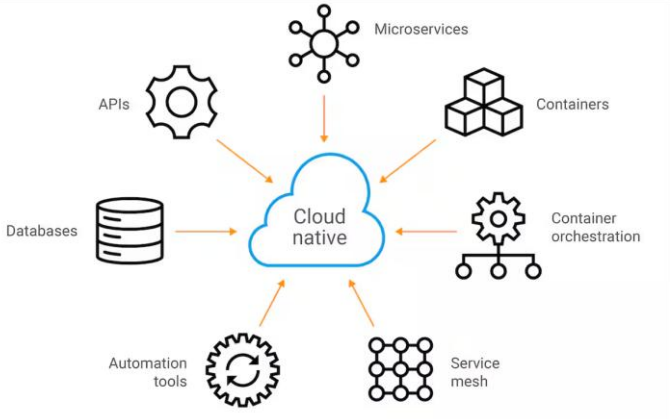


Streaming Analytics & Cloud-Native AI

Real-Time Insights and Elastic Infrastructure

- **Streaming Analytics (Kafka + Spark/Flink with Online ML):**
  - **Purpose:** Applies ML models to data "in motion" for immediate predictions. This is crucial for use cases requiring low latency.
  - **Big Data Example:** Real-time fraud detection systems that must score millions of transactions per second.
- **Cloud-Native AI (AWS Sagemaker, Google Vertex AI, Azure ML):**
  - **Purpose:** These managed platforms handle the underlying infrastructure, allowing teams to focus on building models.
  - **Key Big Data Feature: Autoscaling**. They automatically provision and de-provision compute resources based on workload demands, optimizing both performance and cost.



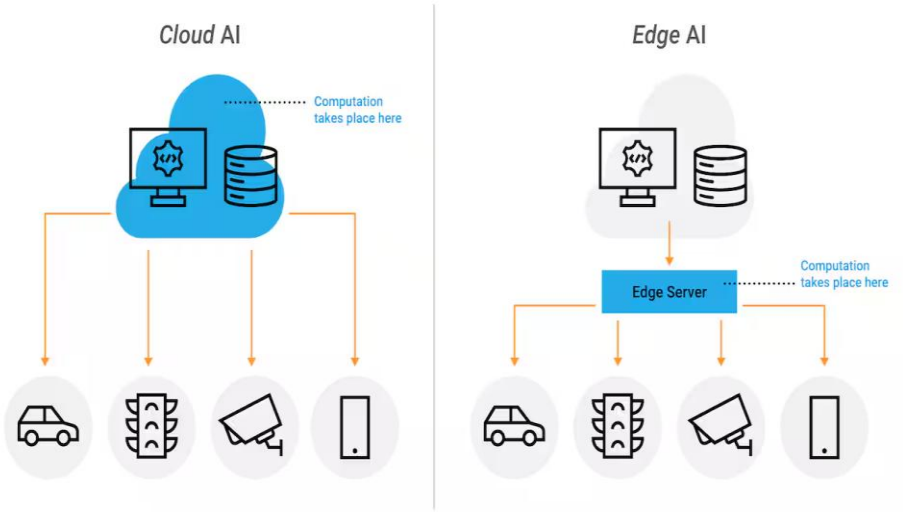


Hybrid & Edge AI

Decentralized & Efficient Big Data Processing

Sending all raw data to a central cloud is often too slow and expensive. This approach solves that problem.

- **Edge AI: Real-Time Local Processing**
    a. **What:** AI models run directly on the data source (e.g., IoT sensors, cameras, vehicles).
    b. **Why:** Enables instant decisions, saves network bandwidth, and keeps sensitive data private.
- **Hybrid AI: Combining Edge + Cloud**
    a. **What:** A strategy where the Edge handles immediate tasks, and the Cloud manages the big picture.
    b. **How it Works:** The **Edge** spots an anomaly in real-time. The **Cloud** receives these summaries from all devices to analyze global trends and deploy smarter models back to the edge.
- **Core Idea:** Deploying smaller, optimized ML/DL models directly onto IoT sensors, cameras, and other edge devices.
- **Big Data Relevance:** Enables real-time, decentralized processing. For example, a smart camera can perform object detection locally and only send an alert to the central system if an anomaly is found. This reduces latency, saves network bandwidth, and enhances



Challenges: Scalability, Interpretability & Privacy

Core Technical and Ethical Hurdles

- **Scalability:** The constant engineering challenge of efficiently training billion-parameter models on petabyte-scale datasets and serving them with low latency.
- **Interpretability (XAI):** Moving beyond prediction accuracy to understand *why* a model makes a certain decision. This is non-negotiable for building trust in critical domains like finance and healthcare.
- **Privacy-Preserving Analytics:** Developing and deploying techniques like **Federated Learning** (training on decentralized data) and **Differential Privacy** (adding statistical noise) to extract insights without compromising individual privacy.

Challenges: Energy Efficiency & Data Bias

Sustainability and Fairness in Big Data

- **Energy Efficiency (Green AI):** Training large foundation models consumes immense amounts of electricity. Research in Green AI focuses on creating more efficient algorithms, hardware, and training methods to make large-scale analytics more sustainable.
- **Data Imbalance & Bias:** Big data is a reflection of the real world, including its historical biases. A major challenge is to develop techniques to detect and mitigate these biases to ensure that AI systems are fair, equitable, and representative.