DEEP LEARNING

Lecture

Introduction

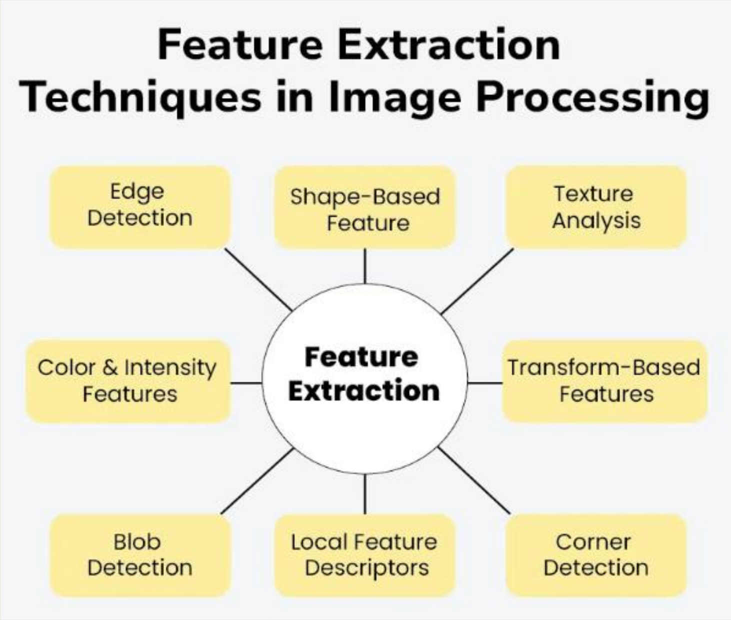
Instructor: Zafar Iqbal

Agenda

- What is Deep Learning?
- Difference between AI, ML and DL
- History, Key components of DL
- Structure, Applications, Challenges
- Frameworks, Mathematics

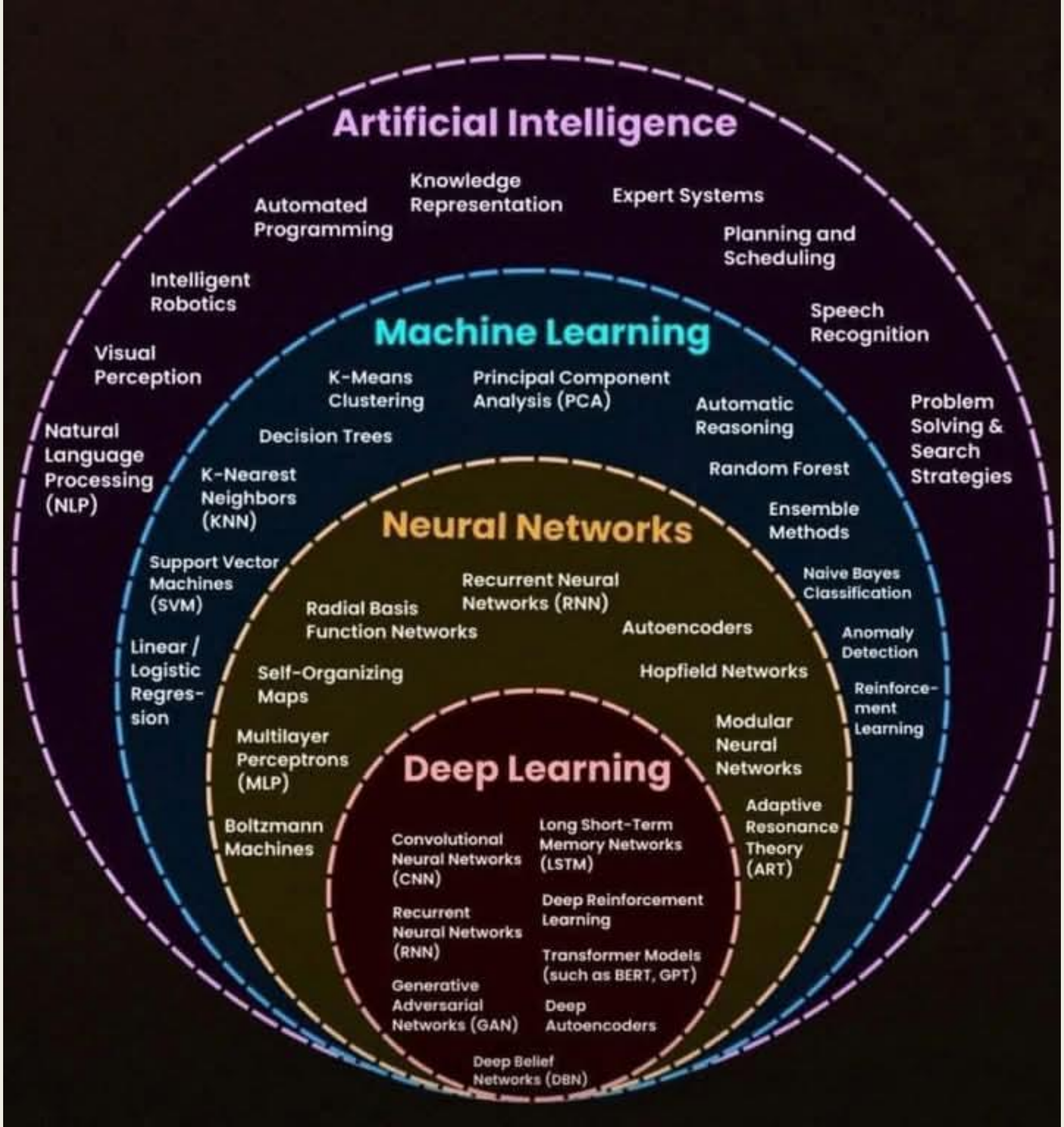
What is Deep Learning?

- A subset of machine learning based on artificial neural networks
- Inspired by the structure and function of the human brain
- Learns hierarchical data representations
- Automatically extracts features from raw data
- Enables machines to perform tasks like humans (vision, language, speech)



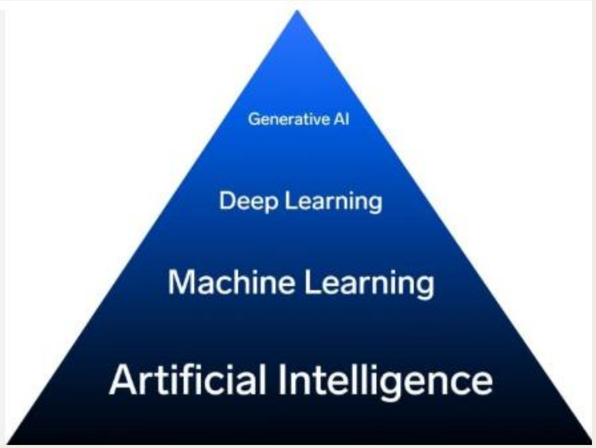
What is Deep Learning

- Neural networks have become the defining model of deep learning due to their power and scalability.
- They are composed of neurons, each of which performs a simple computation.
- The true power of a neural network comes from the complexity of the connections between these neurons.
- These complex connections enable the network to model complicated patterns and relationships in data.


Artificial Intelligence
Automated Programming
Knowledge Representation
Expert Systems
Planning and Scheduling
Intelligent Robotics
Speech Recognition
Visual Perception
Problem Solving & Search Strategies
Natural Language Processing (NLP)
Machine Learning
K-Means Clustering
Principal Component Analysis (PCA)
Automatic Reasoning
Decision Trees
Random Forest
K-Nearest Neighbors (KNN)
Ensemble Methods
Support Vector Machines (SVM)
Naive Bayes Classification
Anomaly Detection
Reinforce-ment Learning
Linear / Logistic Regres-sion
Neural Networks
Recurrent Neural Networks (RNN)
Radial Basis Function Networks
Autoencoders
Self-Organizing Maps
Hopfield Networks
Modular Neural Networks
Multilayer Perceptrons (MLP)
Adaptive Resonance Theory (ART)
Boltzmann Machines
Deep Learning
Convolutional Neural Networks (CNN)
Long Short-Term Memory Networks (LSTM)
Recurrent Neural Networks (RNN)
Deep Reinforcement Learning
Transformer Models (such as BERT, GPT)
Generative Adversarial Networks (GAN)
Deep Autoencoders
Deep Belief Networks (DBN)

AI vs ML vs DL

Level	Description	Example
AI	Broad goal: make machines intelligent	Chatbots, self-driving cars
ML	Algorithms that learn from data	Decision Trees, SVMs
DL	Neural networks with multiple layers	CNNs, RNNs, Transformers



Why Deep Learning?

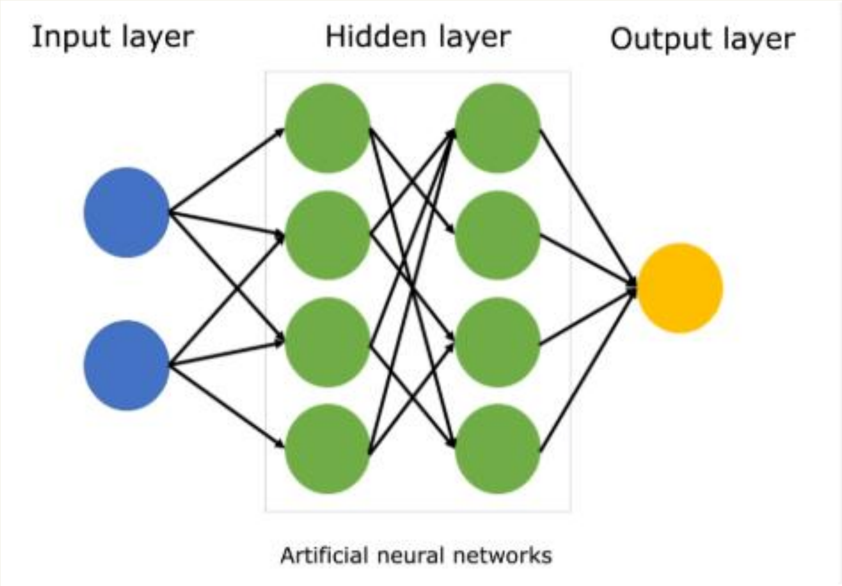
- Handles complex, high-dimensional data
- Requires less manual feature engineering
- Scales well with big data and GPUs
- Achieves state-of-the-art performance in multiple domains

History of deep learning

- 1943: McCulloch-Pitts neuron model
- 1986: Backpropagation algorithm introduced
- 2006: Deep Belief Networks by Hinton et al.
- 2012: AlexNet wins ImageNet, sparking deep learning boom
- 2020s: Transformers and large-scale models dominate

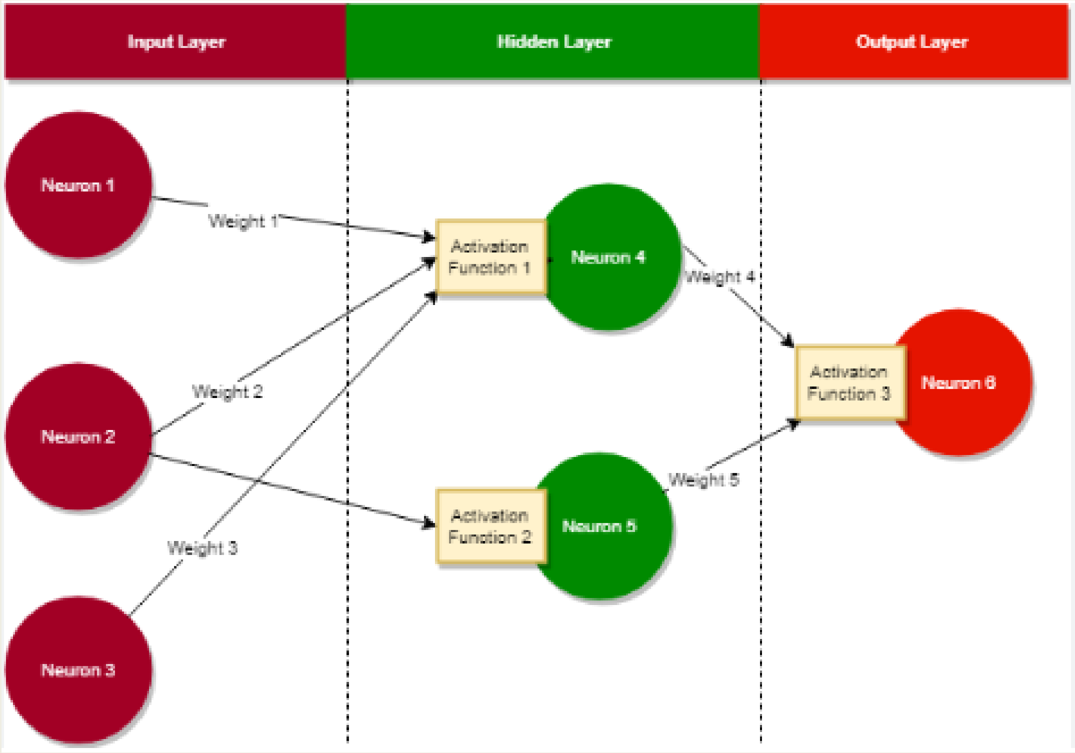
Key Components of Deep Learning

- **Neural Networks:** Core computation structure
- **Data:** Large datasets for learning patterns
- **Computation Power:** GPUs/TPUs for acceleration
- **Algorithms:** Optimization (SGD, Adam), activation functions, loss functions



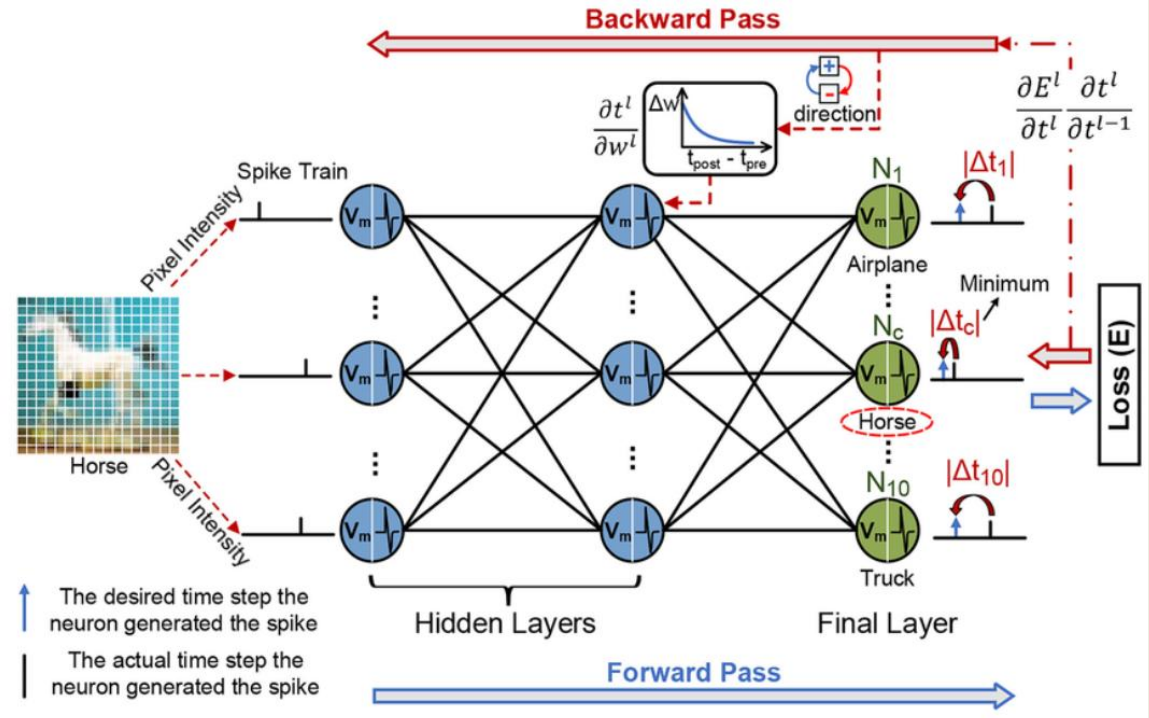
Structure of a Neural Network

- Input layer
- Hidden layers
- Output layer
- Weights and biases
- Activation functions



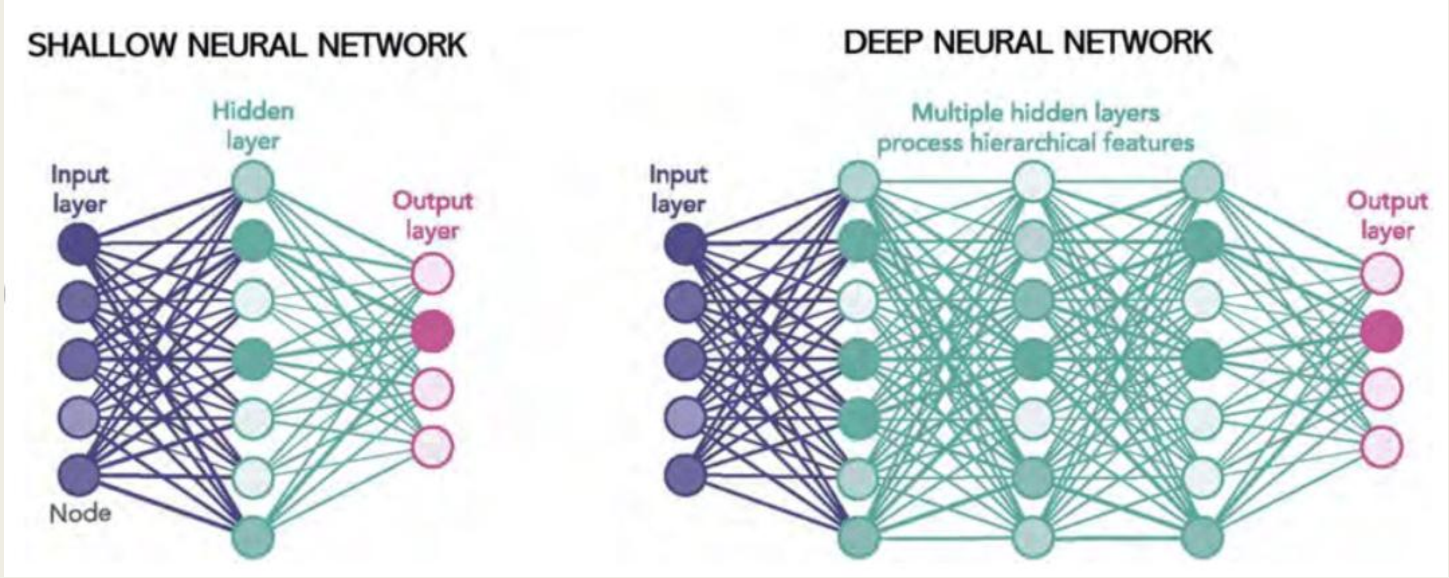
How Neural Networks Learn

- Forward pass: computes predictions
- Loss function: measures prediction error
- Backpropagation: updates weights via gradients
- Optimization loop continues until convergence



Deep vs Shallow Networks

Feature	Shallow	Deep
Layers	1–2 hidden layers	Many layers
Features	Hand-crafted	Learned automatically
Data Need	Small	Large
Accuracy	Limited	High (if data-rich)

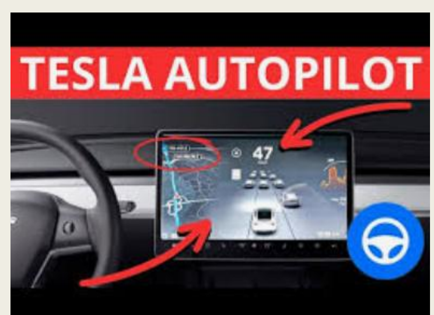


Common Applications

- **Computer Vision:** Object detection, facial recognition
- **Natural Language Processing:** Chatbots, translation
- **Speech Recognition:** Voice assistants
- **Healthcare:** Disease diagnosis, image analysis
- **Autonomous Systems:** Self-driving cars, robotics

Real-World Examples

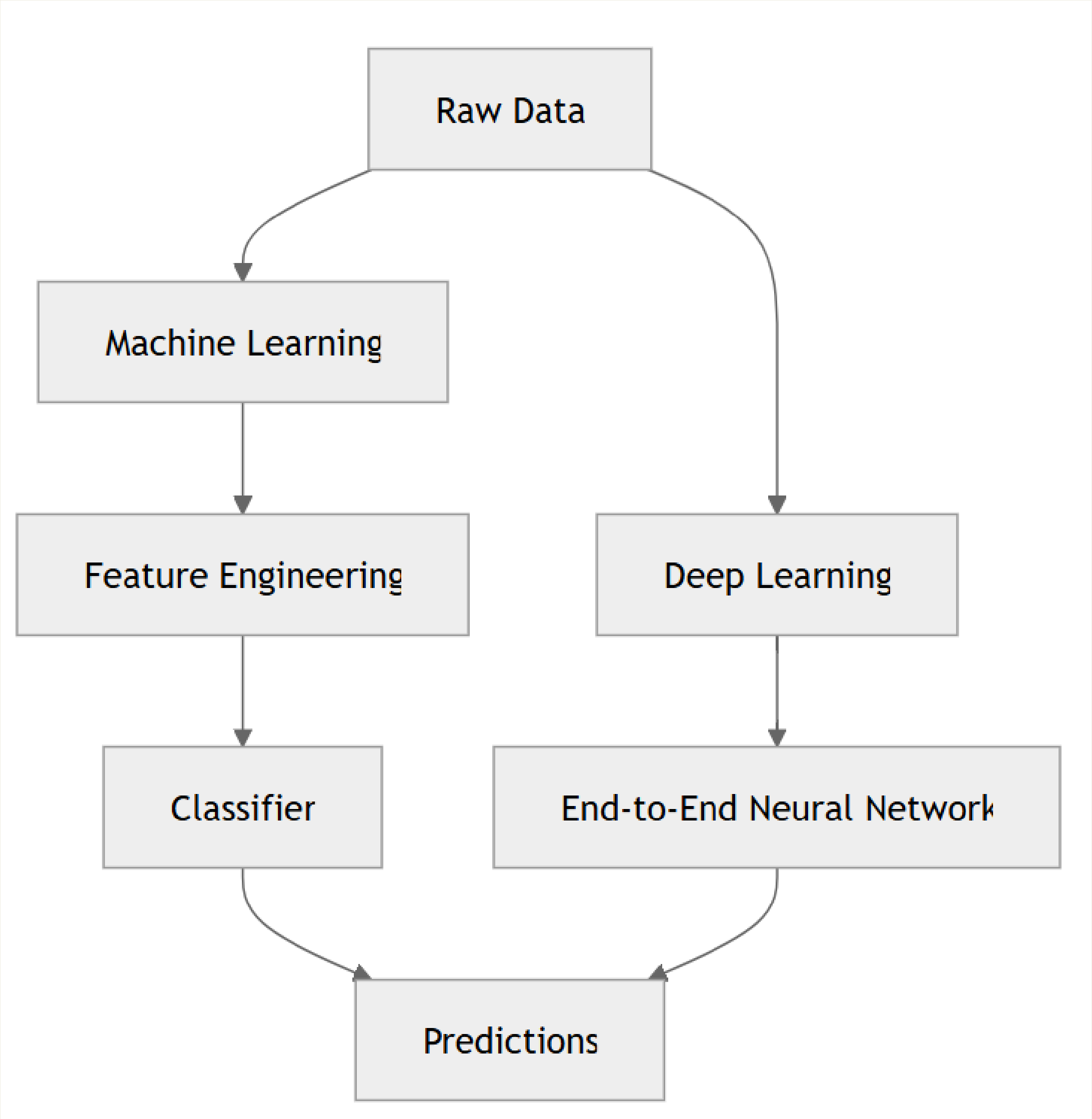
- Google Translate (Sequence-to-sequence models)
- ChatGPT (Transformer models)
- Tesla Autopilot (CNNs for vision)
- DeepMind AlphaGo (Reinforcement learning)



Challenges in Deep Learning

- Data requirements (large labeled datasets)
- Computational cost (training time & GPU needs)
- Interpretability (black-box nature)
- Bias and fairness issues
- Overfitting

Machine Learning vs Deep Learning



Popular Deep Learning Frameworks

- **TensorFlow** (Google)
- **PyTorch** (Meta)
- **Keras** (High-level API)
- **JAX, MXNet, ONNX**
  Open-source, GPU-accelerated, and widely supported

Typical Deep Learning Workflow

- Data collection and preprocessing
- Model design (architecture)
- Training (optimize weights)
- Evaluation (metrics, validation)
- Deployment and monitoring

Deep Learning Ecosystem

- **Hardware:** GPUs, TPUs
- **Software:** Libraries and frameworks
- **Data:** Public datasets (ImageNet, COCO, etc.)
- **Community:** Research papers, GitHub repos

Why Use GPUs?

- Parallel processing for matrix operations
- Speeds up training by 10–100x compared to CPUs
- Cloud options: AWS, Google Cloud, Azure [Image: GPU vs. CPU performance comparison]

Linear Algebra Basics

- Vectors: Represent data points or weights
- Matrices: Represent transformations or connections
- Tensors: Generalize vectors/matrices to higher dimensions
- Operations: Dot product, matrix multiplication Equation: $y = Wx + b$

Calculus for Optimization

- Gradients: Measure rate of change of loss function
- Partial Derivatives: Optimize weights in neural networks
- Gradient Descent: Minimize loss iteratively Equation:
- $w \leftarrow w - \eta * \partial L/\partial w$

Probability and Statistics

- Probability: Model uncertainty (e.g., softmax outputs)
- Expectations: Used in loss functions
- Distributions: Gaussian, Bernoulli for data modeling Equation: $P(y|x) = \exp(w_y * x) / \Sigma(\exp(w_i * x))$ (Softmax)