Artificial Intelligence: A Comprehensive Reference

Table of Contents

Foundational Concepts and Historical Evolution

Key Figures

The roots of AI span decades of research. Among modern pioneers, Yoshua Bengio, Geoffrey Hinton, and Yann LeCun are widely credited for building the foundations of deep learning. The ACM cited these three as "recipients of the 2018 ACM A.M. Turing Award for conceptual and engineering breakthroughs that have made deep neural networks a critical component of computing."

Together, they introduced theories and experiments that demonstrated the power of multilayer neural networks. For example, Hinton's work in the 1980s–90s looked to the human brain for inspiration, advocating for neural nets that mimic cognitive processing.

Other early AI luminaries include Alan Turing (who posited the Turing Test in 1950) and Marvin Minsky (a founder of the MIT AI Lab and co-author of *Perceptrons* in 1969). These figures coined key ideas and shaped AI's trajectory over time.

Foundational Papers in AI, Machine Learning, and Neural Networks

Classic publications established the field. McCulloch and Pitts' 1943 "Logical Calculus of the Ideas Immanent in Nervous Activity" proposed artificial neurons. In 1950, Alan Turing's "Computing Machinery and Intelligence" introduced the question of machine "thinking." Frank Rosenblatt's 1957 paper on the **perceptron** introduced a simple two-layer neural network for pattern recognition.

However, Minsky and Papert's 1969 book *Perceptrons* famously showed single-layer nets could not solve certain tasks, which contributed to a decline in neural-net research. A key revival came with Rumelhart, Hinton & Williams (1986), who described **backpropagation** for multi-layer networks. Notably, an earlier work by Bryson & Ho (1969) first introduced the algorithm.

Other seminal works include the Expert Systems of the 1970s (e.g. DENDRAL, MYCIN), and Alex Krizhevsky et al.'s 2012 paper on a deep **convolutional neural network (CNN)** that dominated ImageNet image classification. In NLP, papers like Devlin et al.'s BERT (2018) and Vaswani et al.'s *Attention Is All You Need* (2017) introduced transformative architectures and concepts (the Transformer) that underpin modern language models.

Generative Adversarial Networks (GANs) were introduced by Goodfellow et al. in 2014, framing generation as a game between two networks. Each of these works is widely cited and marks advances in theory and practice.

Timeline of Major Breakthroughs

AI's history has been punctuated by waves of progress. The **1956 Dartmouth Workshop** (coordinated by John McCarthy, Marvin Minsky, et al.) is often called AI's birth.

Early successes included Samuel's checkers program (1950s) and programs like ELIZA (1966). The field then endured "AI winters" (1970s–80s) when limitations became apparent.

A renaissance came in the 1980s–90s with renewed interest in neural networks and expert systems. A timeline of key milestones would note:

- **1997** – IBM's Deep Blue beats world chess champion
- **2012** – Deep CNN (AlexNet) dramatically improves image recognition
- **2016** – DeepMind's AlphaGo defeats Go champion

The late 2010s saw breakthroughs in NLP (e.g. BERT, GPT) and in 2023–2025 we have seen **GPT-4** (OpenAI) and other multimodal models pushed to human-level benchmarks.

Throughout, AI periodically spurts ahead after paradigm shifts.

Tiers of AI

Understanding AI requires distinguishing categories. **AI (Artificial Intelligence)** is the broad goal of machines performing tasks that require human-like intelligence (e.g. perception, reasoning, planning).

Within AI, **Machine Learning (ML)** is the subset of methods that learn from data. IBM notes that "Machine learning is a subset of AI that allows for optimization" – ML models improve their predictions by minimizing error on data.

Within ML, **Deep Learning (DL)** refers to neural networks with many layers. In essence, DL is "a subfield of machine learning" where neural network architectures (often with more than three layers) automatically learn complex features.

Finally, **Neural Networks (NNs)** are specific architectures of connected nodes (neurons). NNs form the backbone of DL; CNNs, RNNs, and Transformers are all types of neural networks.

- **AI vs. ML:** AI includes any approach (symbolic logic, expert systems, ML); ML specifically uses statistical learning from data.
- **ML vs. DL:** ML includes linear models, decision trees, etc. DL uses deep neural nets and can ingest unstructured data (images, text) end-to-end, often requiring much more data.
- **Neural Networks:** A neural network can be shallow (few layers) or deep. When a network has many hidden layers (depth), we typically call it a deep neural network or deep learning model. Classic ML might use hand-crafted features, whereas DL "automates much of the feature extraction" process.

IBM's blog captures this hierarchy concisely: *"AI is the overarching system. Machine learning is a subset of AI. Deep learning is a subfield of machine learning, and neural networks make up the backbone of deep learning algorithms."*

Another useful breakdown classifies AI by capability: **Narrow AI (ANI)** performs specific tasks, while hypothetical **AGI/ASI** would match or exceed human versatility. For practical use, almost all current AI systems are narrow AI built with ML/DL techniques.

Types of Neural Networks

Modern AI uses several core architectures:

- **Feedforward Networks / MLP**: The simplest neural nets (multilayer perceptrons) with input, hidden, and output layers. These were popular early DL models for generic tasks.

- **Convolutional Neural Networks (CNNs)**: Designed for data with spatial structure (images, video). CNNs use convolution and pooling layers to detect local patterns. They exploit the fact that images have nearby pixel correlations. CNNs dramatically reduced parameters versus a fully connected net and excel at vision tasks.

- **Recurrent Neural Networks (RNNs)**: Used for sequential or time-series data (text, speech, time series). RNNs pass information along a sequence by having connections that loop. In theory they can "remember" previous inputs. In practice, vanilla RNNs struggled with long-range dependencies, leading to variants like LSTM and GRU.

- **Transformer Networks**: Introduced in "Attention Is All You Need", Transformers eschew recurrence entirely and rely on attention mechanisms. A Transformer encoder-decoder uses attention layers to weigh relationships between all input tokens. Transformers form the backbone of nearly all modern NLP (and vision) models.

- **Generative Adversarial Networks (GANs)**: Proposed in 2014, where two networks – a generator and a discriminator – compete in a minimax game. The generator creates samples from random noise, and the discriminator learns to distinguish real versus generated samples. GANs revolutionized image and data generation.

Other architectures include Autoencoders, Variational Autoencoders (VAEs), Graph Neural Networks (GNNs), and Mixture-of-Experts (MoE) networks.

Training Techniques and Loss Functions

Neural networks learn by optimizing a loss function over training data using gradient-based methods. Common loss functions include:

- Mean squared error (MSE) for regression
- Cross-entropy (log loss) for classification

Optimization is typically done with variants of stochastic gradient descent (SGD). Many systems use adaptive optimizers like Adam or RMSProp.

Regularization techniques:

- Dropout: Randomly deactivating neurons during training
- Batch normalization: Stabilizing training
- Weight decay (L2 regularization)
- Data augmentation

Hardware and Processing Architecture

CPU vs GPU vs TPU

- **CPU**: General-purpose processor optimized for low-latency tasks
- **GPU**: Contains thousands of smaller cores for parallel numeric computation
- **TPU**: ASIC designed for tensor-heavy ML workloads

CUDA Cores vs Tensor Cores

- **CUDA cores**: General-purpose ALUs for typical parallel instructions
- **Tensor cores**: Specialized units designed to accelerate mixed-precision matrix operations

Infrastructure for Large-Scale Training

Training state-of-the-art models often requires clusters of GPUs or TPUs with high-performance interconnects and fast storage.

Multimodal and Cutting-Edge Models

Recent years have seen multimodal AI models that handle text, images, audio, and more:

- **GPT-4**: Large-scale transformer-based model that accepts both text and image inputs

- **Gemini**: Google's multimodal models using transformers with Mixture-of-Experts architecture
- **Claude**: Anthropic's LLMs designed for high safety and utility
- Other models: DALL·E, Imagen, Stable Diffusion, etc.

NLP and Large Language Models (LLMs)

Natural Language Processing Fundamentals

Modern NLP leverages deep learning with techniques like:

- Tokenization
- Embedding
- Transformer architectures

Large Language Models: Training and Function

Training recipe:

1. Pretrain Transformer to predict tokens
2. Fine-tuning on specific tasks
3. Often use Reinforcement Learning from Human Feedback (RLHF)

Applications and Challenges

Applications:

- Chatbots
- Writing assistants
- Code generation
- Language translation

Challenges:

- Hallucination
- Bias and fairness
- Scalability
- Ethical use and security

Training Techniques and Loss Functions

Knowledge Distillation

Knowledge distillation is a model compression technique where a smaller student model learns to mimic a larger teacher model.

- **Student-Teacher Models**:
    - The teacher model generates output distributions (soft labels) on training data

  - The student model is trained to match these outputs using a combined loss function
- **Advantages**:
  - Reduces model size and computation time
  - Preserves performance
  - Improves generalization
- **Applications**:
  - Model compression
  - Knowledge transfer between architectures
  - Improving ensemble or multi-task models

Inference-Time Parameters

Parameters that control model outputs during generation:

- **Temperature**: In AI models, particularly in generative AI like language models, temperature controls the randomness and creativity of the output. Higher temperatures lead to more diverse and potentially creative outputs, while lower temperatures result in more predictable and focused outputs.
  - Scales logits before softmax
  - `>1`: Flatter distribution (more random)
  - `<1`: Sharper distribution (more deterministic)
- **Top-k Sampling**:
  - Considers only top k highest-probability tokens
  - Improves coherence while allowing variety
- **Top-p (Nucleus) Sampling**:
  - Chooses smallest set of tokens exceeding probability p
  - Dynamic approach adapts to distribution shape
- **Beam Search**:
  - Keeps multiple partial hypotheses during generation
  - Produces more coherent outputs but computationally expensive

Training-Time Hyperparameters

Parameters affecting model learning:

Parameter	Description	Impact
Learning Rate	Step size in gradient descent	High: Faster but unstable Low: Slower but stable
Batch Size	Examples per weight update	Large: Smoother gradients Small: More noise

Parameter	Description	Impact
Epochs	Complete dataset passes	More: Better learning Risk of overfitting
Weight Decay	L2 regularization term	Prevents overfitting
Gradient Clipping	Limits gradient magnitude	Prevents exploding gradients
Dropout Rate	Fraction of disabled units	Improves generalization

Model Architecture Parameters

Parameters defining network structure:

- **Number of Layers**:
    - More layers capture complex features
    - Risk of vanishing gradients
- **Hidden Units per Layer**:
    - More units increase capacity
    - Risk of overfitting
- **Activation Functions**:
    - ReLU, tanh, sigmoid
    - Enable learning complex patterns
- **Attention Heads (Transformers)**:
    - Parallel attention mechanisms
    - More heads capture diverse relationships
- **Positional Encodings**:
    - Inject sequence order information
    - Can be fixed or learned

Loss Functions

- **Cross-Entropy Loss**: Classification tasks
- **Mean Squared Error (MSE)**: Regression tasks
- **Hinge Loss**: Max-margin classifiers (e.g., SVM)

Optimizers

Optimizer	Characteristics	Best For
SGD	Basic gradient descent	Simple problems
Adam	Combines momentum + RMSProp	Sparse/noisy gradients
RMSProp	Adaptive learning rates	Non-stationary objectives

Ethical Considerations and AI Safety

Bias in Training Data

AI systems can perpetuate or amplify bias present in their training data. Mitigation involves:

- Curating diverse datasets
- Fairness-aware learning algorithms
- Post-processing outputs

Explainability and Interpretability

Techniques include:

- LIME or SHAP
- Visualizing activation maps
- Attention heatmaps

Responsible AI Frameworks

Principles include:

- Fairness
- Transparency
- Accountability
- Privacy
- Human oversight

AI Safety: Alignment and Reward Hacking

Focuses on:

- Matching AI goals with human values
- Preventing reward hacking
- Building fail-safes

Use Cases and Real-World Applications

Healthcare

- Medical imaging
- Drug discovery
- Personalized medicine

Finance

- Algorithmic trading
- Fraud detection

- Risk modeling

Robotics and Autonomous Systems

- Self-driving cars
- Industrial robots
- Drones

Marketing and Recommendations

- Recommendation engines
- Content personalization
- Marketing analytics

Creative AI

- Text generation
- Image generation
- Music composition

AI/ML Resources that I Found Helpful to Make This Document

Start here to learn about the basics of AI and Machine learning. This gives a nice foundation and vocabulary for the following resources. Here is an IBM explanation that tells the differences between all the terms mentioned in the previous article.

Note that I view all of the different 'types' of AI as tiers because AI is really a blanket term, but the world tends to group things into 'AI' in general. This document puts the tiers in order of complexity and intelligence/connections as I see them after months of research.

Tiers of 'AI'

Tier 1: Artificial Intelligence

IBM's Explanation of AI

Tier 2: Machine Learning

IBM's Explanation of Machine Learning

Tier 3: Deep Learning

IBM's Explanation of Deep Learning

Deep learning applications and uses

Tier 4: Computer Vision

AWS Explanation

IBM Explanation

The Parts

Neural Networks

MIT's Explanation of Neural Networks (surface level)

IBM's Neural Network Explanation (in depth)

IBM Developer VERY deep dive into learning in neural networks

Types of Neural Networks

IBM Convolutional Neural Networks (CNN's)

IBM Recurrent Neural Networks (RNN's)

Turing explanation of Transformer Models

Transformer network step by step

NLP VS. LLM

Comprehensive overview of NLP V LLM

Uses & Applications of NLP & LLM

IBM's Explanation of NLP

Model Training

Oden Technologies on Model training

Model training on hardware & differences in hardware

Hardware

CPU V. GPU

TRG Data Centers

IBM's Explanation of the computational differences for AI

Hardware Requirements

What Hardware is needed (overview)

Scalability, Parts, Hardware, explanations

More complex explanation

Architecture and Processing

How GPU-Based AI Processing Works

NVIDIA GPU explanation

Basic guide to hardware and architecture for AI

AI Hardware Explanation

Intel's explanation of AI hardware

Processing efficiency from MIT study

CUDA Cores VS Tensor Cores

CUDA Explanation 1

CUDA Explanation 2

Tensor Cores Explanation

Tensor V CUDA

Videos

Transformers (great visualizations!)

More transformers Explanation

Transformers Again

Recurrent Neural Networks, Transformers, and Attention

Convolutional Neural Networks

Neural Networks

Neural Network Architecture

The 7 Types of AI

All Machine learning models

Types of AI (Deep Dive)

Why so many Foundation models?

How to pick a foundation model

Most Important ML Algorithms

Multimodal Models

How LLM's Work

5 Minute Neural Network explanation

What are transformers?

How AI learns

How to train your own model