Lecture 1a
Introduction to Deep Learning

Contents

1 Introduction to Deep Learning

1.1 Definition

Deep Learning is a subfield of Artificial Intelligence (AI) and Machine Learning (ML) that uses artificial neural networks with multiple layers to model complex patterns in data.

In deep learning, algorithms tend to analyze data within a logical system continuously. This helps deep learning construct patterns that relate to the human brain essential in making conclusions and decisions.

The roots of deep learning lie not only in computer science, but also in *psychology (associationism)*, *neuroscience (connectionism, synaptic models)*, and *computer architecture (Von Neumann machines)*.

1.1.1 Artificial Intelligence and Its Branches

Artificial intelligence(AI) makes machines capable of performing tasks that normally require human intelligence, which includes reasoning, planning, language understanding, vision, learning, creativity, etc.

Artificial Intelligence (AI) is a broad field with several major branches, each focusing on different aspects of intelligence and computation. The key branches include:

- **Symbolic AI**
  - Rule-based systems, logic, expert systems.
  - Example: IF–THEN rules in early medical expert systems.
- **Search & Planning**
  - Algorithms for decision-making, such as A* and minimax in games.
  - Pathfinding in robotics and problem-solving domains.
- **Knowledge Representation & Reasoning (KRR)**
  - Ontologies, semantic networks, and formal logic.
  - Probabilistic reasoning frameworks such as Bayesian networks.
- **Machine Learning (ML)**
  - Algorithms that learn patterns from data instead of being explicitly programmed.
  - Sub-branches:
    - **Supervised Learning**: classification, regression.
    - **Unsupervised Learning**: clustering, dimensionality reduction.
    - **Reinforcement Learning**: agents learn by trial and error with feedback.
- **Natural Language Processing (NLP)**
  - Enabling machines to understand and generate human language.

  - Examples: GPT, Mistral, RoBERTa.
- **Computer Vision**
  - Understanding and analyzing images and videos.
  - Applications: object detection, facial recognition, medical imaging.
- **Robotics & Perception**
  - Embodied AI systems that sense and act in the physical world.
  - Integration of perception, planning, and control.

1.2 Theory of Associationism

Associationism is a psychological theory, which states that all knowledge and thought are built from associations — simple links formed between ideas, events, or sensations.

When two events occur together often enough, the mind associates them. Later, one event automatically brings the other to mind.

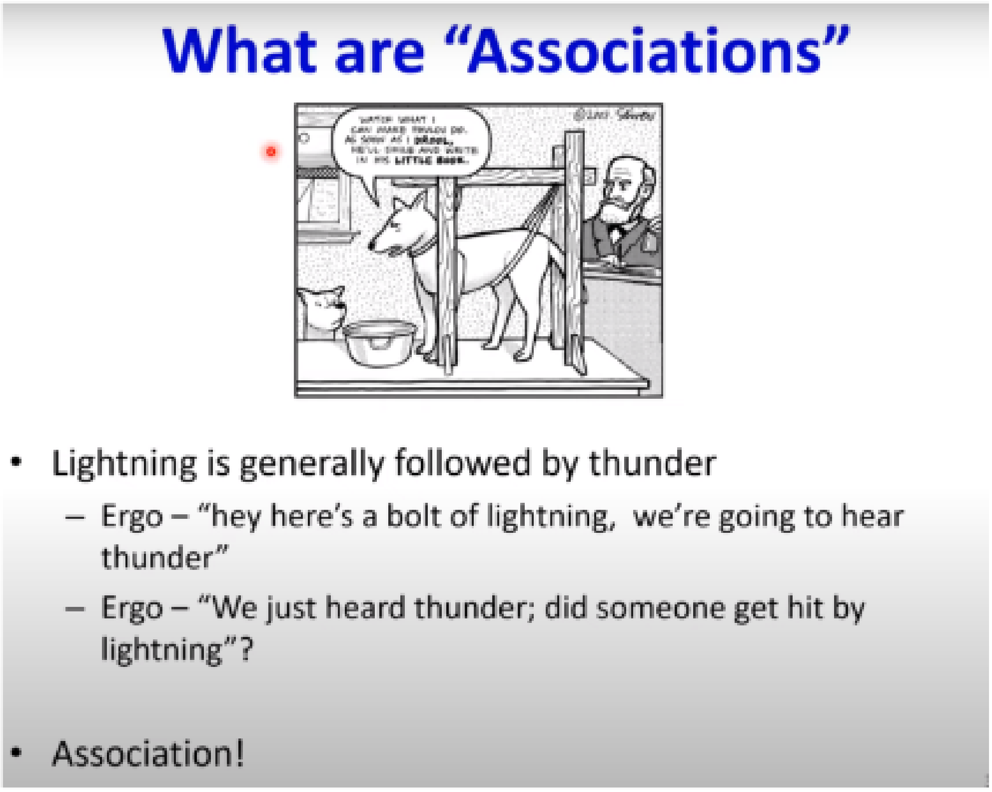

Figure 1: Associations in classical conditioning: repeated pairing (e.g., bell → food) forms a mental link, leading to conditioned responses.

Key Principles of Associationism:

- **Contiguity:** Ideas or events that occur together in time/space become linked.
- **Frequency:** The more often two things occur together, the stronger the association.
- **Similarity/Contrast:** Ideas that resemble each other, or are opposites, also form associations.

Relevance to AI and Neural Networks:

- Early researchers borrowed from associationism: machines can learn by associating input patterns with output responses.
- Example: Show a computer many cat pictures → it associates certain features (whiskers, ears) with the label "cat".

1.3 Theory of Connectionism

Connectionism is a theory inspired by neuroscience, which suggests that mental processes arise from interconnected networks of simple processing units (neurons). Knowledge is not stored in single symbols or rules, but in the *pattern of connections and weights* across a network.

How it Works:

- A network consists of simple units (artificial "neurons") connected by weighted links.
- Each unit computes a simple function (e.g., weighted sum + activation).
- Learning occurs by adjusting the connection weights, so the network can capture input-output relationships.

Mathematical Representation:

$$y = f\left(\sum_i w_i x_i + b\right)$$

where x_i are inputs, w_i are weights, b is bias, and $f(\cdot)$ is a non-linear activation function (e.g., sigmoid, ReLU).

Key Principles of Connectionism:

- **Distributed Knowledge:** Information is stored in weights across the network, not in a single unit.
- **Parallel Processing:** Many neurons process information simultaneously.
- **Learning by Adjustment:** Training adapts weights through algorithms like gradient descent and backpropagation.

1.4 Computer & the Von Neumann Architecture

1.5 Computer & the Von Neumann Architecture

Core Idea:
The Von Neumann architecture is the standard model for how modern computers are structured. In this model:

- The **processor** (CPU) is separate from the **memory**.
- Memory holds both *data* and *programs*.
- The processor reads instructions and data from memory to perform tasks.
- The processor is independent of the program and the data: you can load a different program into memory at any time, and the same CPU will execute it.

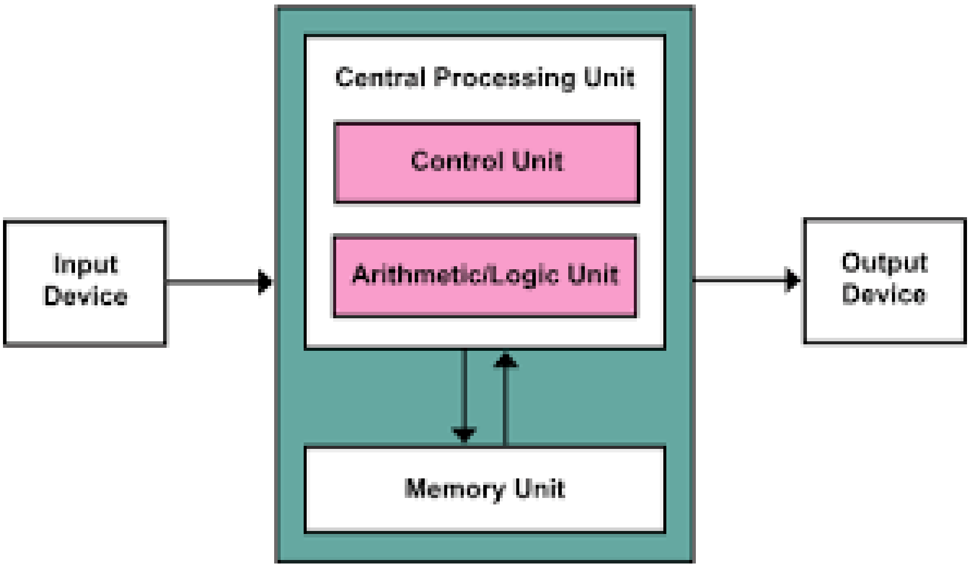

Figure 2: The Von Neumann architecture.

Contrast with Neural Networks:

- In a connectionist machine (like an artificial neural network), there is **no separation** between processor and memory.
- The "program" is encoded directly in the **connections (weights)** between neurons.
- To change the program, you must change the structure or weights of the network itself.
- This is fundamentally different from the Von Neumann architecture, where changing the program only requires loading new instructions into memory.

1.6 Human Brain

Biological Neural Networks:
The human brain is a massively interconnected web of biological neurons conducting complex distributions of electrical signals. Anatomically, the brain can be distinguished

into several major divisions such as the **frontal lobe**, **Parietal lobe**, **Occipital lobe** and **cerebellum**.

Each division can be further broken down into specialized areas and regions according to the functions they perform and the underlying structure of their neural networks.

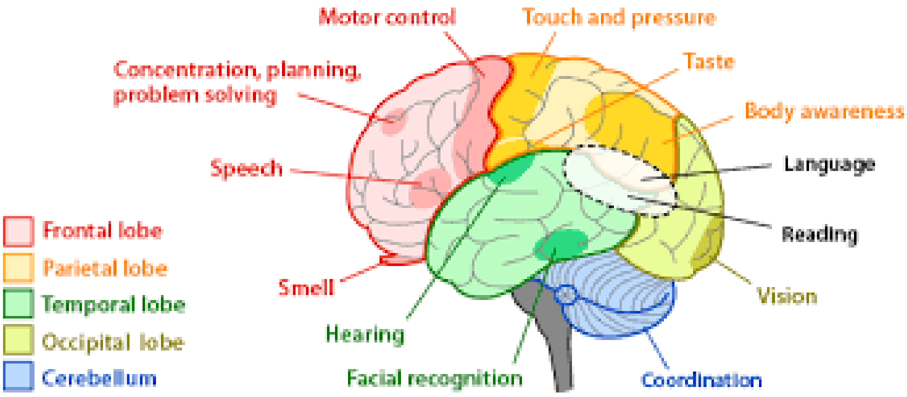

Figure 3: Major anatomical divisions of the human brain.

Neurons:
The foundational unit of the human brain is the **neuron**. The general brain architecture contains many networks of interconnected neurons, each utilizing biochemical reactions to send, receive, and process information.

A neuron consists of:

- **Dendrites:** Input structures that receive electrical signals or impulses from other neurons.
- **Cell Body (Soma):** Integrates the incoming signals and determines whether the neuron will fire.
- **Axon:** Transmits the output signal away from the soma to other neurons.
- **Synapse:** A small gap where the axon terminal of one neuron connects to the dendrite of another neuron. Signals are transmitted across the synapse chemically (via neurotransmitters) or electrically.

When the summation of inputs received by the dendrites exceeds a certain threshold, the neuron fires an **action potential**, which travels along the axon to the next neuron. This process underpins all brain activity.

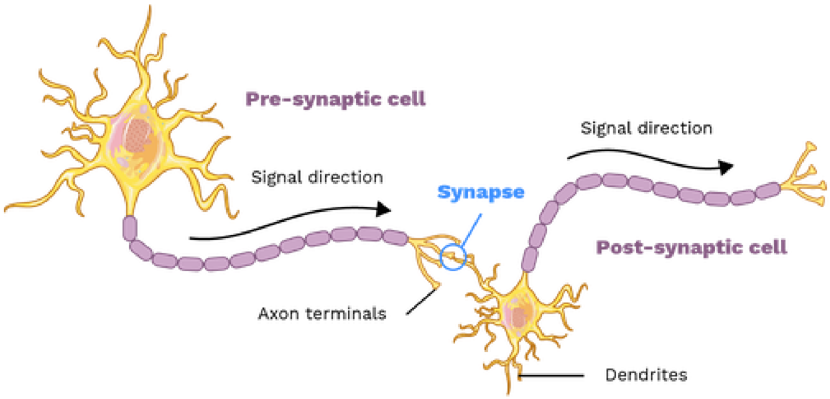

Figure 4: Anatomy of a biological neuron

1.6.1 Boolean Gates in the Synaptic Model

Core Idea:
The earliest artificial neuron models (McCulloch & Pitts, 1943) demonstrated that networks of simple neurons could compute basic logical operations. By treating neuron firing as binary (1 = fires, 0 = does not fire), neurons can act like **Boolean logic gates**.

How It Works:

- Inputs x_i are binary values (0 or 1).
- Each input has a weight w_i that represents synaptic strength.
- The neuron computes a weighted sum:

$$y = f\left(\sum_i w_i x_i - \theta\right)$$

  where θ is the threshold and $f(\cdot)$ is a step activation function.
- If the sum $\geq \theta$, the neuron outputs 1 (fires); otherwise it outputs 0.

Examples of Boolean Gates:

- **AND Gate:** Two inputs x_1, x_2 with weights $w_1 = w_2 = 1$, threshold $\theta = 2$. Fires only if both inputs are 1.

$$y = f(x_1 + x_2 - 2)$$

- **OR Gate:** Two inputs x_1, x_2 with weights $w_1 = w_2 = 1$, threshold $\theta = 1$. Fires if at least one input is 1.

$$y = f(x_1 + x_2 - 1)$$

- **NOT Gate:** One input x_1 with weight $w_1 = -1$, threshold $\theta = 0$. Fires when $x_1 = 0$.

$$y = f(-x_1)$$