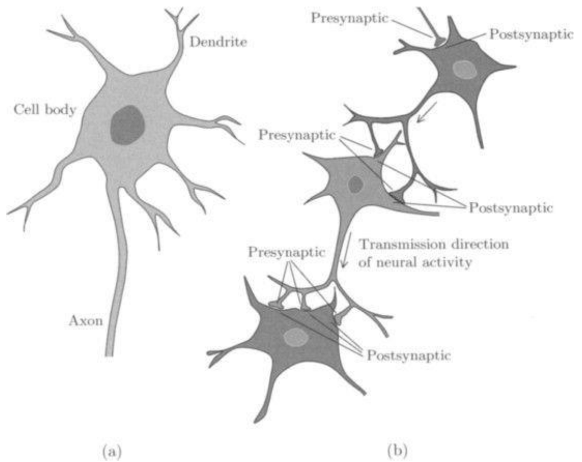
History of AI

Prof. Laxmidhar Behera

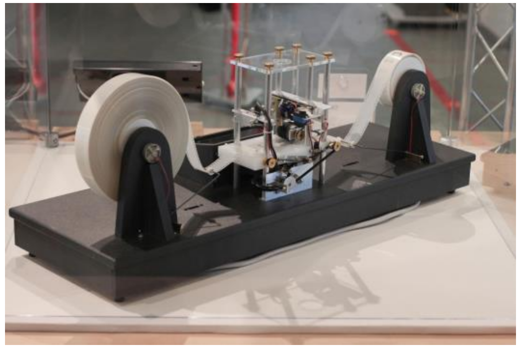
Director, IIT Mandi

Birth of artificial intelligence (1930-40) – Connecting Biology and Computation



- **1930-1940 -** The Human Brain as an Electrical Network

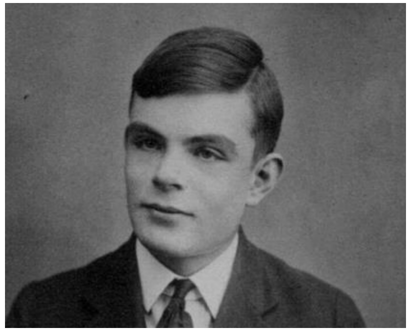
Research in neurology - Brain was an electrical network of neurons that fired in ***all-or-nothing pulses***. Neurons will either transmit an impulse over the synapse to the next neuron completely or not at all.



- **1936 -** The Birth of Digital Computation

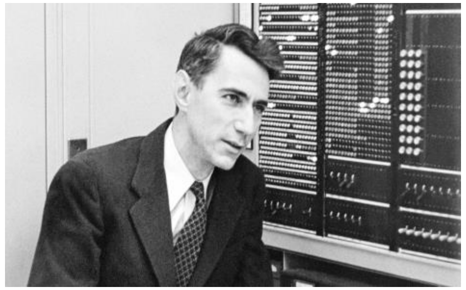
Alan Turing - Computation could be performed using a simple machine that operated using only two states: 0 and 1

Turing's "machine" could theoretically perform any **calculation** that a computer can perform.



- **1940s** : Information Theory and Digital Signals

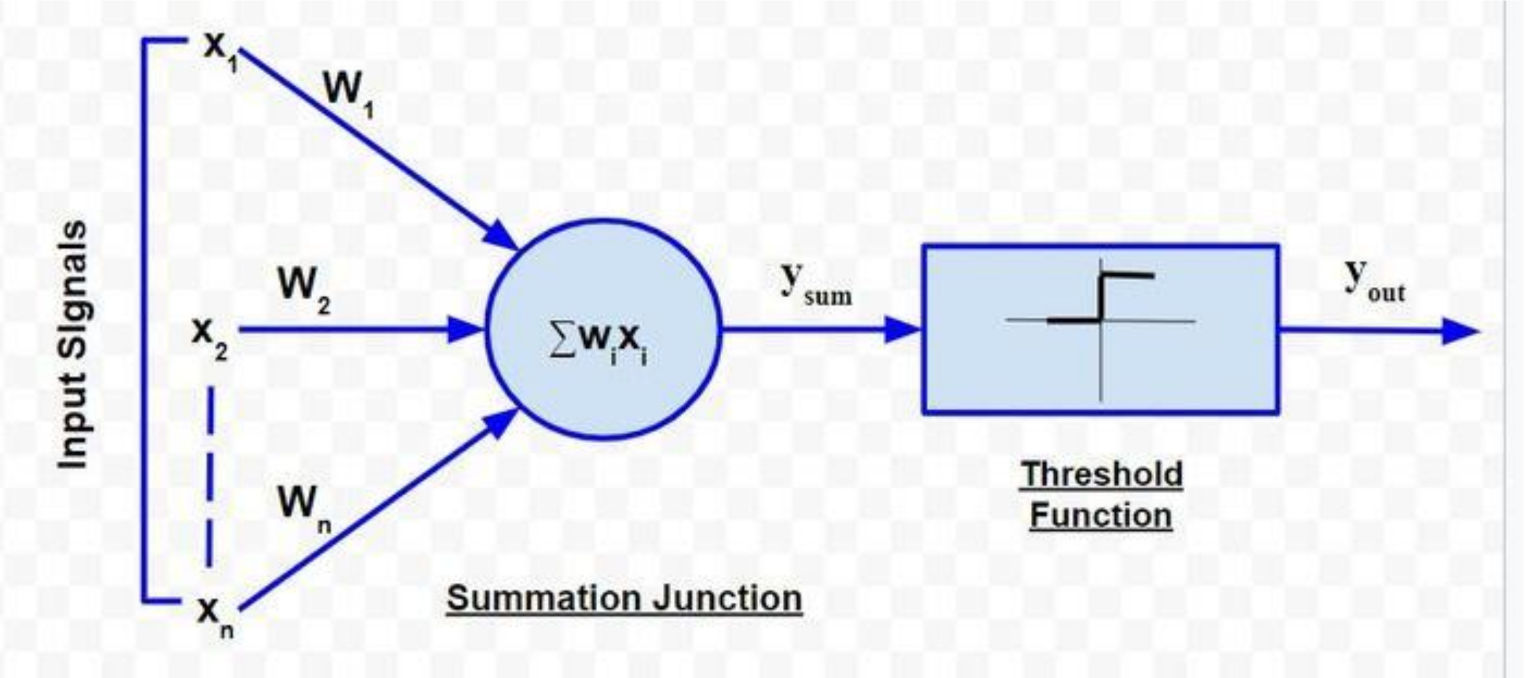
Claude Shannon - Developed *information theory,* how information could be represented and transmitted using digital signals, which, again, are based on 'on' and 'off' states (0s and 1s).

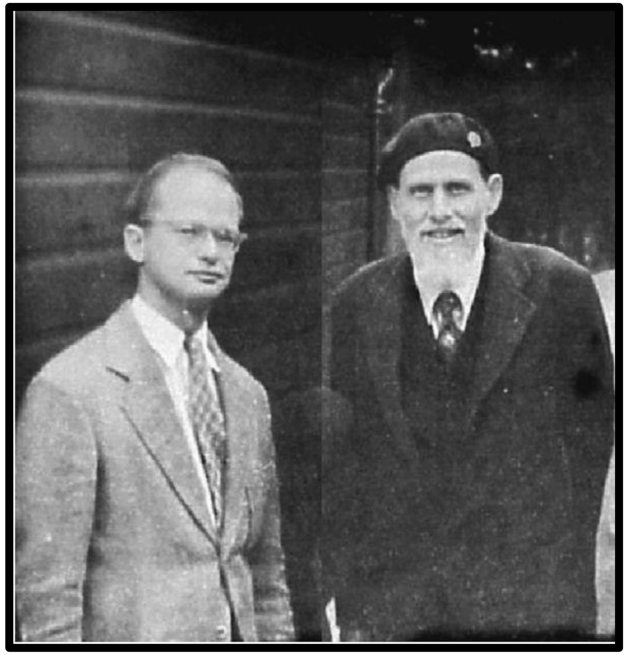


- The close relationship between these ideas suggested that it might be possible to construct an **"electronic brain".**

Birth of artificial intelligence (1941-56) - McCulloch and Pitts model

- **1943** - Warren McCulloch and Walter Pitts - A simplified mathematical model of how the neurons in our brains operate





A neuron model that sums binary inputs and outputs a 1 if the sum exceeds a certain threshold value and otherwise outputs a 0.

$$y_{out} = \begin{cases} 1, & y_{sum} \geq \theta \\ 0, & y_{sum} < \theta \end{cases}$$

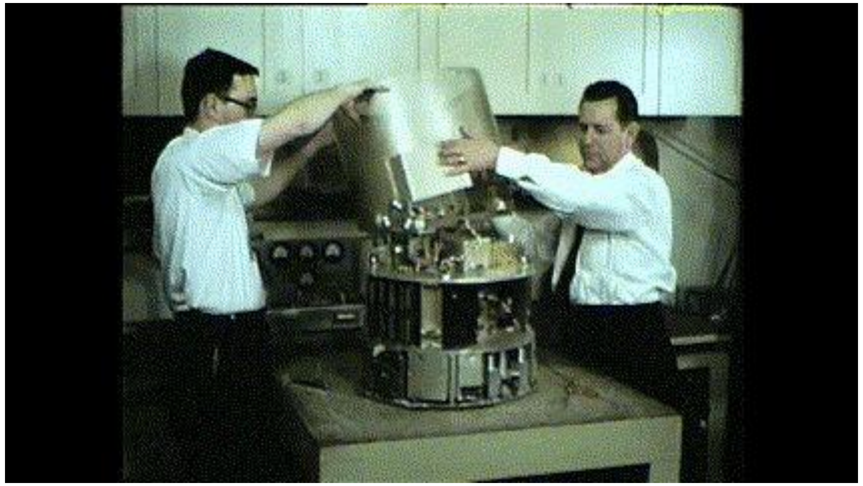
where θ is the threshold

Birth of artificial intelligence (1941-56) - Early Autonomous Robots – Pre-Digital Era

Late 1940s:

Grey Walter's Turtles

- W. Grey Walter, a neurophysiologist, built experimental robots
- To understand how simple biological systems, like the nervous systems of simple animals, could produce complex behaviors
- Instead of using digital computers (which use 0s and 1s), Walter's turtles were controlled by analog circuitry
- They could navigate towards light sources, avoid obstacles



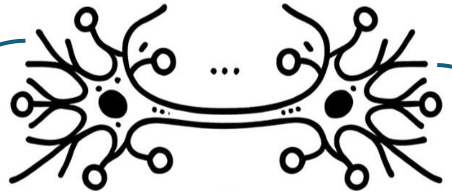
The Hopkins Beast

•Used a combination of sensors (sonar and photocells) and analog circuitry to navigate and recharge itself.

Birth of artificial intelligence (1941-56) – Hebbian learning

- **1949** - Donald Hebb – Hebbain theory -- "Neurons that fire together, wire together" - if two neurons are active simultaneously, the connection between them should be strengthened.

Presynaptic neuron



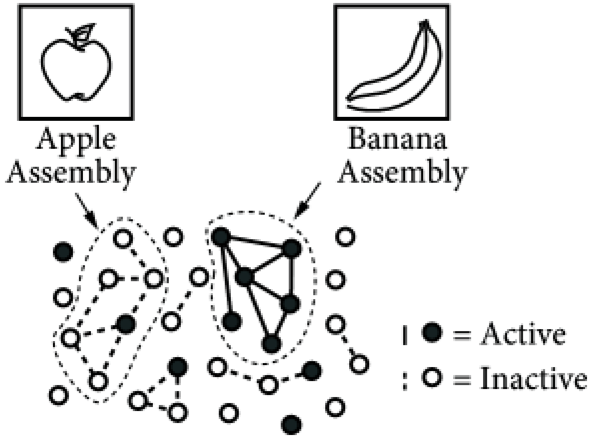
Postsynaptic neuron

$$\Delta w_{ij} \propto x_i \cdot y_j$$

Δw_{ij} is the change in the synaptic weight between the i-th presynaptic neuron and the j-th postsynaptic neuron.

x_i is the activation of the i-th presynaptic neuron.

y_j is the activation of the j-th postsynaptic neuron.



Birth of artificial intelligence (1941-56) – Hebbian learning – An example

Scenario: We have two "neurons" (or feature detectors):

- Neuron A: Detects vertical lines.
- Neuron B: Detects horizontal lines.
- We present the system with several handwritten images of digits.

Example 1: The Digit "1"

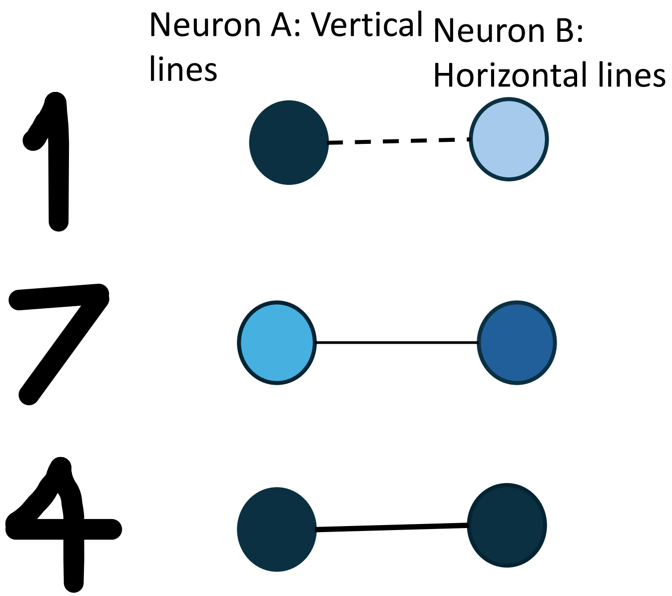
- $x_A = 1, x_B = 0.1 \rightarrow \Delta w_{AB} = \eta * 1 * 0.1 = 0.1\eta$
- The weight is increased slightly.

Example 2: The Digit “7"

- $x_A = 0.7, x_B = 0.8 \rightarrow \Delta w_{AB} = \eta * 0.7 * 0.8 = 0.56\eta$
- The weight is increased slightly more than in the case of the "1“.

Example 3: The Digit “4"

- $x_A = 0.9, x_B = 0.9 \rightarrow \Delta w_{AB} = \eta * 0.9 * 0.9 = 0.81\eta$
- The weight is increased even more.



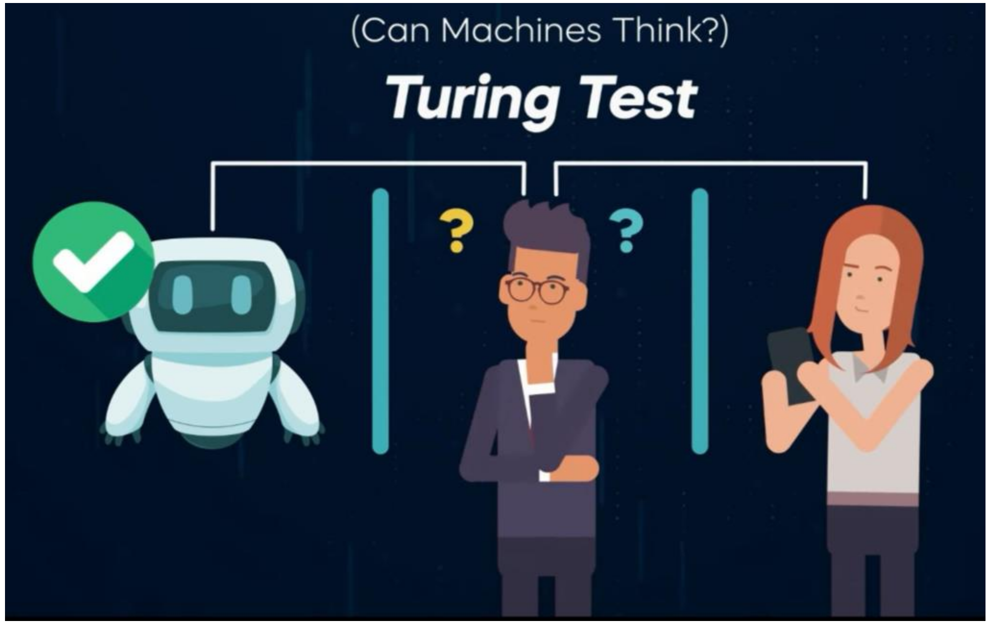
- **Hebbian learning** : Neurons in the visual cortex adapt to prefer specific orientations of lines or edges based on the patterns of visual input they receive during development. This process ***strengthens connections*** between neurons that are ***repeatedly activated together***, helping the brain interpret visual stimuli.

Birth of artificial intelligence (1941-56) - Turing Test (Can machines Think)

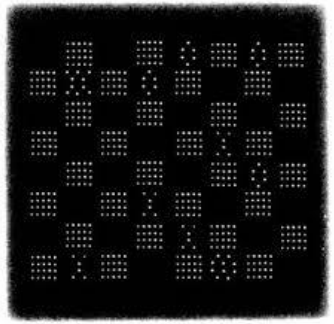
1950: Alan Turing published “Computing Machinery and Intelligence”, proposing the Turing Test to define "thinking machines.

The Turing Test involves a human interrogator engaging in a natural language conversation with one human and one machine—both hidden from view.

If the interrogator cannot reliably distinguish the machine from the human, the machine is considered to have passed the test—human-like intelligence.



Birth of artificial intelligence (1941-56) - First AI Game Programs



1951 – Inspired by the vision of Turing, Strachey and Prinz created the first AI programs for checkers and chess, simulate human-like decision-making using **brute-force search on all the legal moves** and basic decision-making.

1952 - Arthur Samuel developed the first computer learning program for checkers, which improved by analyzing winning moves, marking an early milestone in machine learning.

Birth of artificial intelligence (1941-56) - The Dartmouth Conference

Finally in 1956, the term "artificial intelligence" was coined by John McCharthy at a Dartmouth Conference, and AI became a formal field of study.

Attendees: John McCharthy, Marvin Minsky, Claude Shannon, Nathaniel Rochester, Arthur Samuel, Trenchard More, Ray Solomonoff, Oliver Selfridge, Allen Newell, Herbert Simon.

They agreed to bring together experts interested in neural networks, the theory of computation and automata theory.

To explore if machines could simulate various aspects of human intelligence.

Early Successes in AI (1956–1974) – Perceptron model

- **1958** - Frank Rosenblatt introduced the perceptron, a single-layer neural network that extends the McCulloch-Pitts model by incorporating a learning rule for adjusting weights
- Rosenblatt drew inspiration from Hebb's work because the idea of adjusting synaptic strengths based on neuronal activity was a key biological inspiration for early neural networks
- ***He predicted its future ability to learn and translate languages.***

- For each training input–output pair (x, y), where $x = (x_1, x_2, \ldots, x_n)$

  Compute the activation:

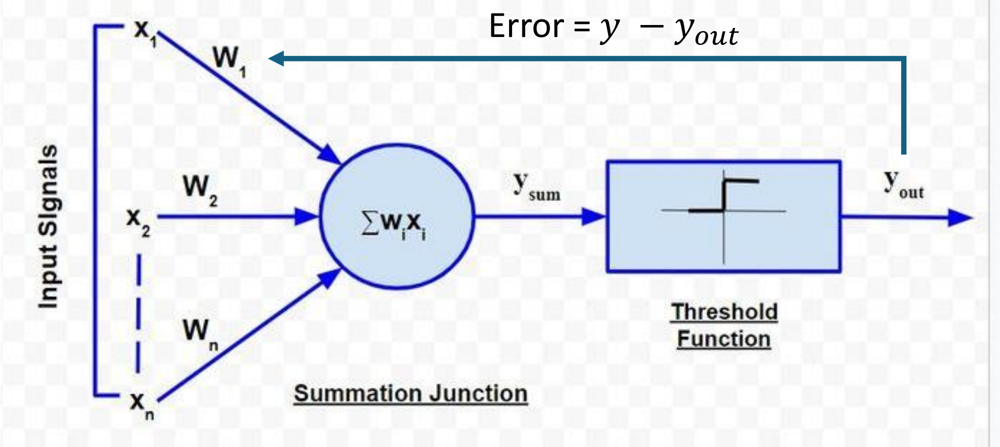
$$y_{sum} = \sum w_i x_i + b,$$

  Apply the decision rule based on threshold:

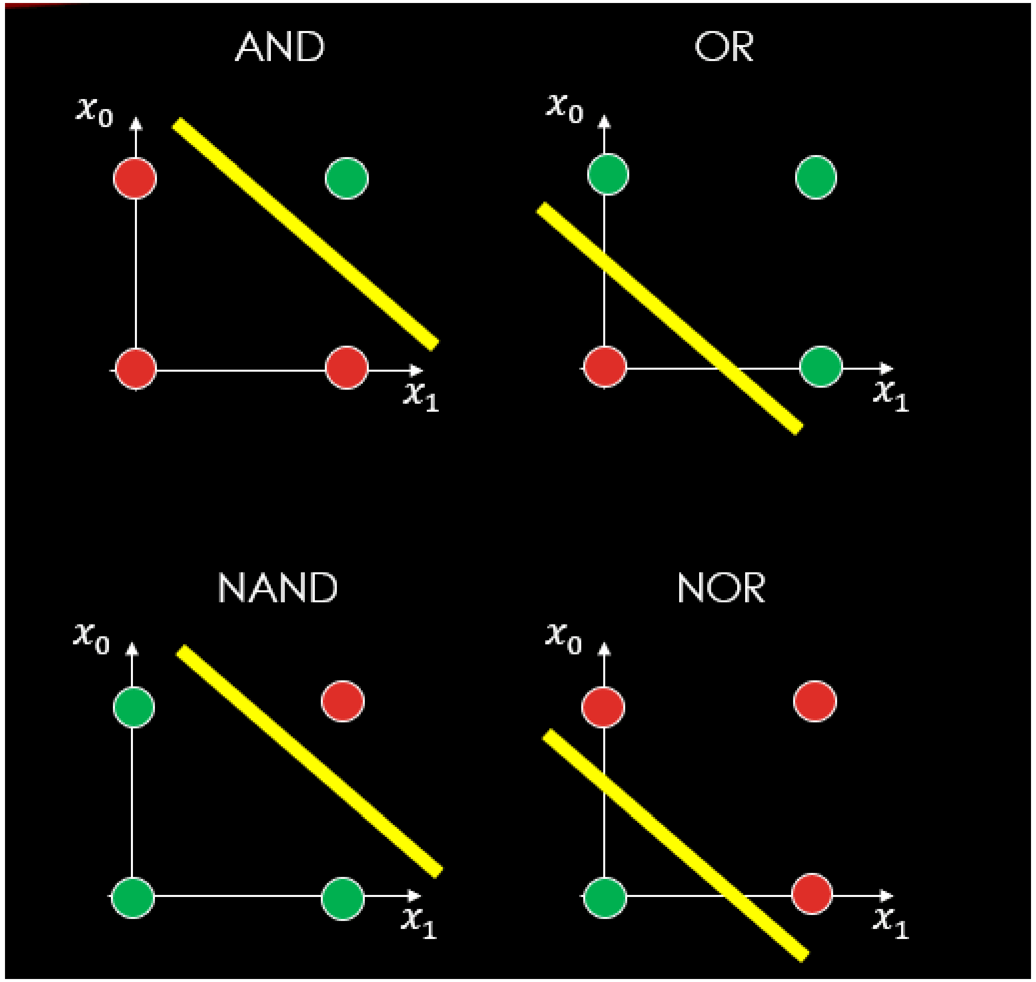
$$y_{out} = \begin{cases} 1, & y_{sum} \geq \theta \\ 0, & y_{sum} < \theta \end{cases}$$

  Update weights:

$$w_i \leftarrow w_i + \eta(y - y_{out})x_i$$

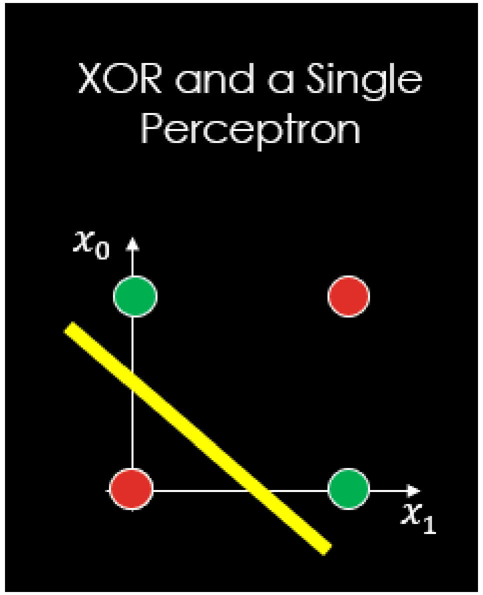


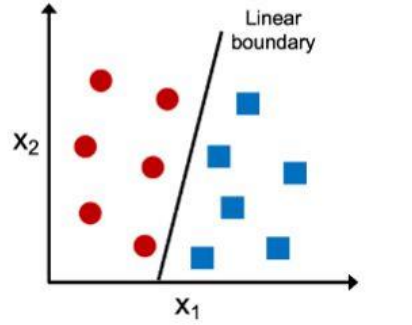
Early Successes in AI (1956–1974) – Perceptron model

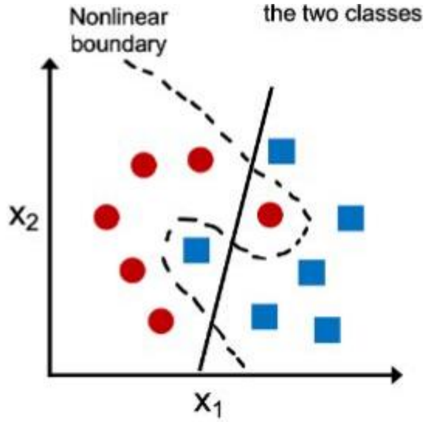


A single perceptron can learn any function, as long as the dataset is linearly separable, like AND, OR, NAND, and NOR!

BUT not for nonlinearly separable such as XOR!!!



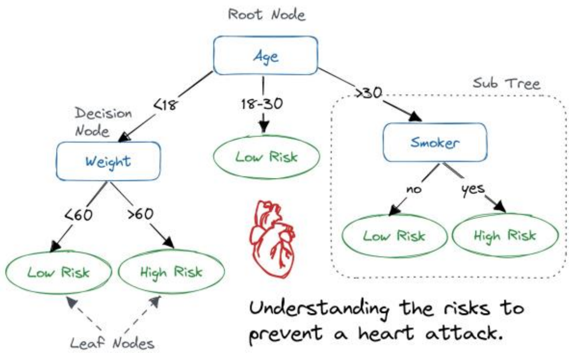


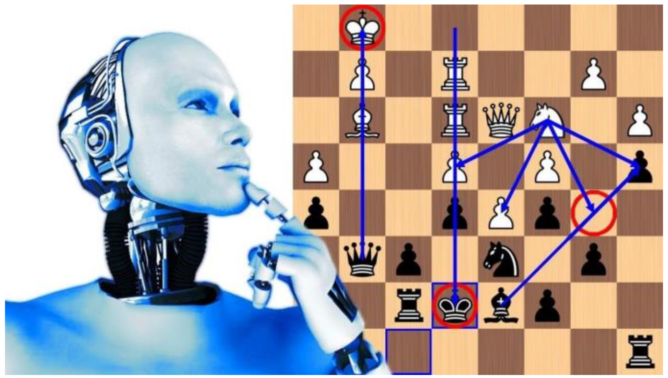


Early Successes in AI (1956–1974) - Early AI Funding and Paradigms

In late 1950s–Early 1960s

- **Increased Funding:** Large-scale funding began for AI research.
- **DARPA's Role:** The US Defense Advanced Research Projects Agency (DARPA), created in response to Sputnik, became a major funding source. This funding supported research at universities like MIT, Carnegie Mellon, and Stanford.





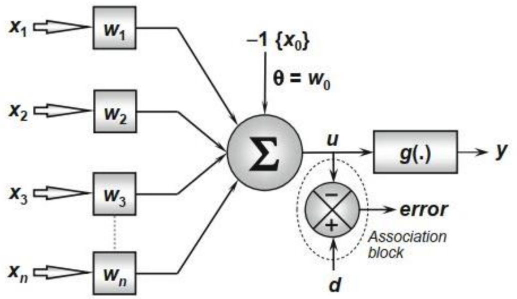
Two Dominant Paradigms Emerged

Symbolic AI (Logic-Driven)

- Represents knowledge with symbols and logical rules.
- Replicates intelligence by manipulating these symbols using predefined rules.
- Examples include theorem proving, chess playing, and grammar correction.

Connectionist AI (Neural Networks)

- Represents knowledge as connections between nodes (neurons) in a network
- Replicates intelligence by learning patterns and relationships in data through adjusting the weights of these connections
- Examples include image recognition, natural language processing, and speech recognition

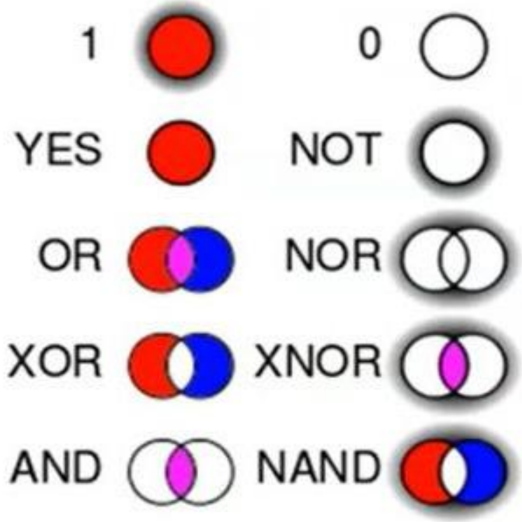


Early Successes in AI (1956–1974): Logic Theorist – Symbolic AI

The Logic Theorist (Simon & Newell, 1956) was an early symbolic AI program designed for automated theorem proving

It employed heuristics (rules of thumb), specifically "means-ends analysis" (comparing the current state to the goal and applying operators to reduce the difference), to efficiently navigate the search space, avoiding brute-force search

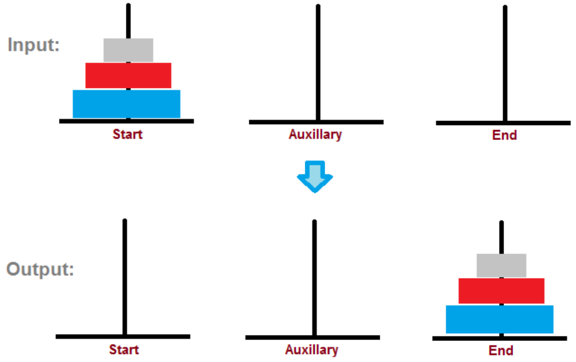
The program successfully proved 38 of the first 52 theorems from Russell and Whitehead's *Principia Mathematica*.

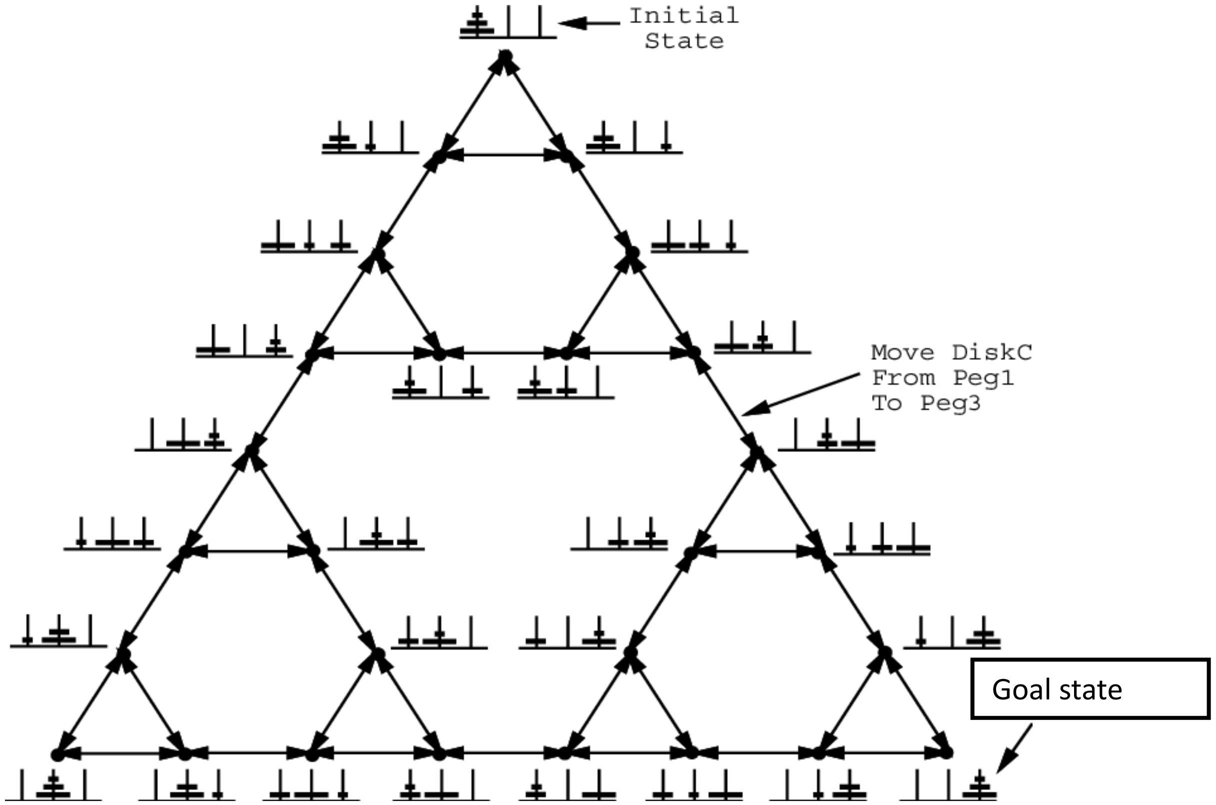


Early Successes in AI (1956–1974): Logic Theorist – An example

Tower of Hanoi Problem:
Move all the disks from the source tower to the destination tower while maintaining their order

- Move one disk at a time.
- A larger disk cannot be placed on top of a smaller disk.





- A ***brute-force approach*** would explore *every possible move* at each step
- The number of possibilities grows ***exponentially*** as the number of disks grow.
- A simple **heuristic**: "Focus on moving the largest unsolved disk by repositioning smaller disks to support it" solves in less number of steps
- Heuristics is a key concept in the history of AI and a ***central contribution of the Logic Theorist***

Early Successes in AI (1956–1974): LISP

LISP (List Processing) programming language:
One of the earliest high-level programming languages.

It introduced concepts like symbolic computation, recursion, and automatic memory management, which were revolutionary at that time.

Task: Add 2 and 3 and store the result

Machine Code

0001: LOAD (load a value into a register - a small storage location in the CPU)
0010: ADD (add two registers)
0011: STORE (store a register's value to memory)
2: 0010
3: 0011

```
0001 0000 0000 0010 ; LOAD 2 into Register 0 (0000 is the register number)
0001 0001 0000 0011 ; LOAD 3 into Register 1 (0001 is the register number)
0010 0010 0000 0001 ; ADD Register 0 and Register 1, store in Register 2 (0010 is the destination register)
0011 0010 0000 1000 ; STORE Register 2 into memory address 8 (1000 is the memory address in binary)
```

Assembly Language

```
LOAD 2, Register1 (Load the value 2 into a register)
LOAD 3, Register2
ADD Register1, Register2, Register3
STORE Register3, 1000 (Store the contents of Register3 into a memory location (let's say address 1000)
```

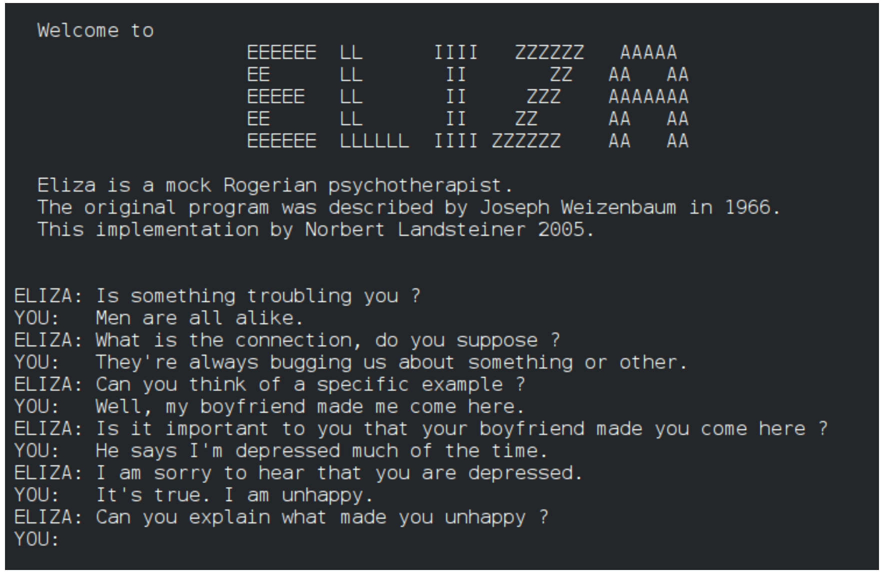
LISP Language

```
(+ 2 3) ; This directly evaluates to 5
 or to store it in a variable: (setf result (+ 2 3))
```

Feature	Low-Level	High-Level
Abstraction	Very low; close to the hardware	High; abstracts away hardware details
Complexity	Complex; requires understanding of CPU architecture	Simple; focuses on the logic of the problem
Readability	Difficult for humans to read and understand	Easy for humans to read and understand
Portability	Not portable; machine-specific	Portable; can run on different machines
Memory Management	Manual; programmer is responsible for memory allocation	Automatic (garbage collection in Lisp)
Speed (Potentially)	Can be very efficient	Might be slightly less efficient in some cases (but the difference is often negligible)

Assembly language was the initial programming method for early microcontrollers (e.g., **Intel 8051, Motorola chips**), followed by **Lisp** for specialized, often AI-related, microcontroller applications.

Early Successes in AI (1956–1974): ELIZA
- Early Attempts at NLP



- First program capable of attempting the Turing test.
- Developed by Joseph Weizenbaum at MIT.
- Simulated a Rogerian psychotherapist.

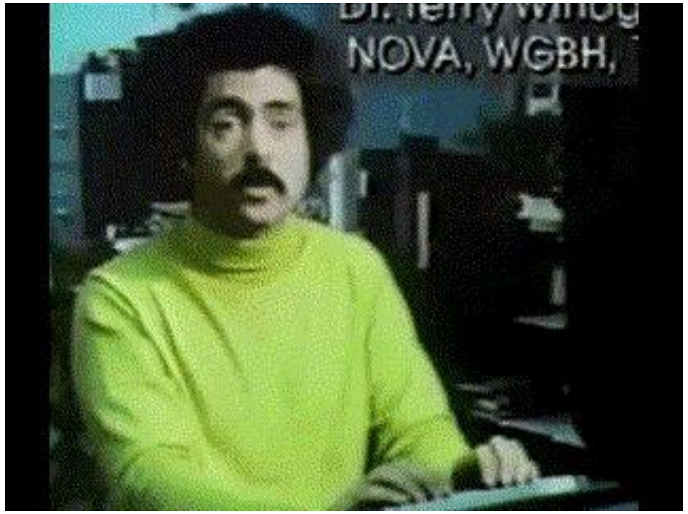
- Used simple pattern matching and substitution to generate responses.
- Examples
  - “I am *” → “Why are you *”
  - “You are *” → “What makes you think I am *”
  - “Everyone *” → “Who is an example?”
- ELIZA lacked any real understanding of language, context, or the world.

Towards MICROWORLDS:

- Since the real world included too many hidden variables for early AI to feasibly interpret, the next best thing was to create AI that could operate within restricted, digital worlds

- In the late 60’s, Marvin Minsky et al. suggested that the artificial intelligence community shift their research to working in ***microworlds.***

Early Successes in AI (1956–1974): SHRDLU: A Major Advance in Natural Language Understanding



- Developed by ***Terry Winograd at MIT***.
- Operated in the blocks world micro-world.
- Showed that computers could understand and act based on natural language within a limited domain.
- **Examples**
  - User: Find a block which is taller than the one you are holding and put it into the box.
  - SHRDLU: BY "IT" I ASSUME YOU MEAN THE BLOCK WHICH IS TALLER THAN THE ONE I AM HOLDING.

  - User: Is there a block which is supported by a pyramid?
  - SHRDLU: YES, THE BLUE BLOCK.

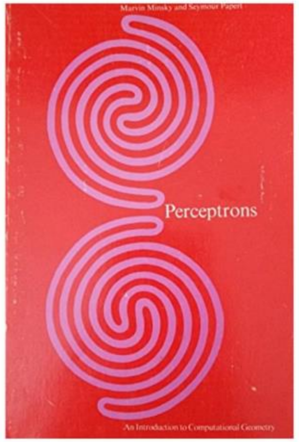
  - User: Did you pick up a pyramid in the past?
  - SHRDLU: YES, I DID.

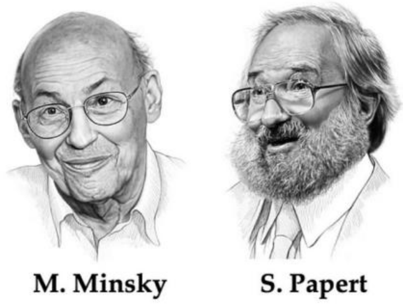
- **Limitations**: Its understanding was limited to the blocks world. It lacked common sense and the ability to generalize to other domains.

Early Successes in AI (1956–1974): Learning Algorithms

1969

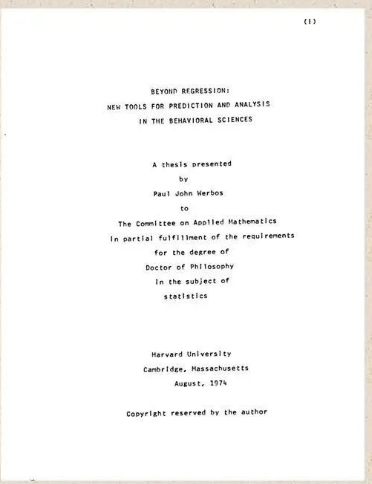
- Minsky and Papert published the book Perceptrons, highlighting limitations of single-layer neural networks, halting connectionist research for a decade, contributing to the first AI winter.
- Example: Unable to represent the XOR (exclusive-or) operation.
- Multi-layer networks were not studied due to lack of learning algorithms.





1974

- Paul Werbos's 1974 Ph.D. thesis introduced **backpropagation, ordered derivatives**, and laid groundwork for **feedforward/feedback networks** and **adaptive dynamic programming**, but these pioneering neural network concepts went ***largely unrecognized*** until the 1980s. His work, though initially overlooked due to computational limitations and prevailing research trends, became foundational to modern deep learning and AI.
- **Backpropagation** provided a method to train multi-layer networks by computing error gradients through layers.



(1)

BEYOND REGRESSION:
NEW TOOLS FOR PREDICTION AND ANALYSIS
IN THE BEHAVIORAL SCIENCES

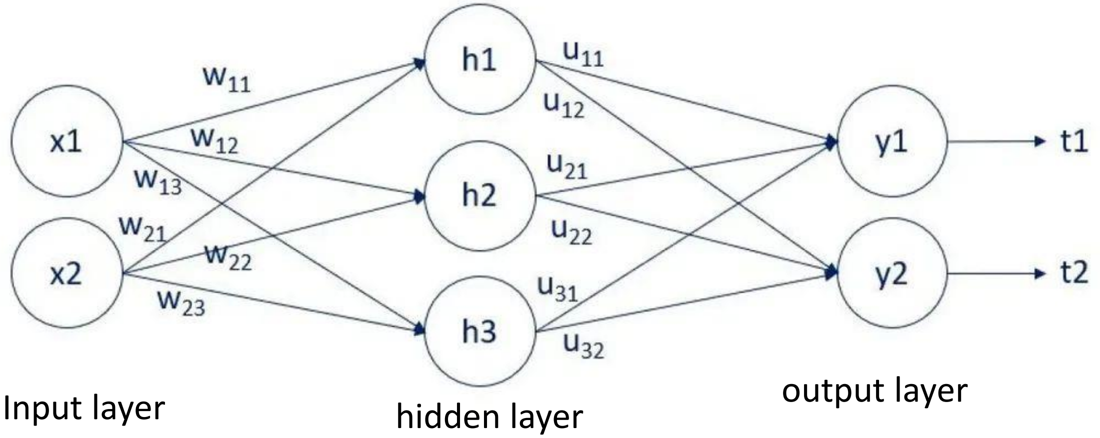
A thesis presented
by
Paul John Werbos
to
The Committee on Applied Mathematics
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of
statistics

Harvard University
Cambridge, Massachusetts
August, 1974

Copyright reserved by the author

Early Successes in AI (1956–1974) - Backpropagation

- **1974 -** Backpropagation, invented by Paul Werbos, gained attention only after the 1986 paper by David Rumelhart, Geoffrey Hinton and Ronald Williams.



Forward propagation is the process of pushing inputs through the net

$$a_j^{(1)} = \sum_i w_{ij} x_i \qquad a_j^{(2)} = \sum_i u_{ij} h_i$$

$$h_j = \sigma(a_j^{(1)}) \qquad y_j = \sigma(a_j^{(2)})$$

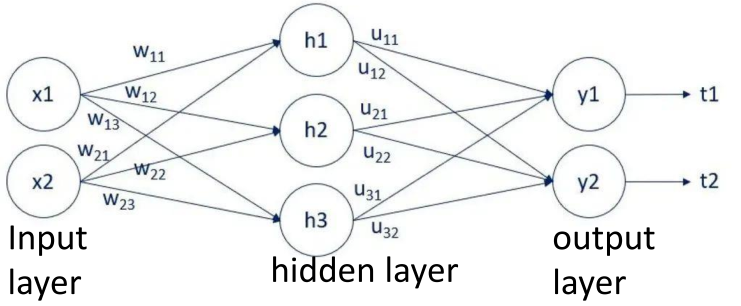
The sigmoid (logistic function) is one of the most common non-linearities

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Early Successes in AI (1956–1974) - Backpropagation

<u>Cost(or Loss) function</u>: It is a function of the difference between estimated (y) and true values (t) for the data

$$L = \frac{1}{2}\sum_i (y_i - t_i)^2$$



<u>Central Idea</u> - Compute the gradient of the loss function. Decrease the loss by iteratively adjusting the weights and biases based on the obtained gradients, and the network gradually learns to make better predictions

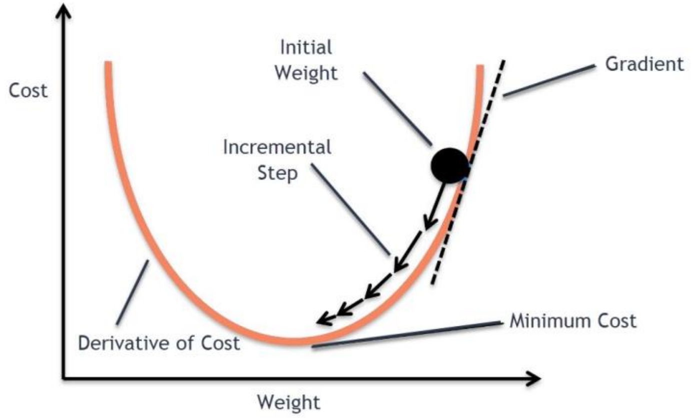
$$\mathbf{u} \leftarrow \mathbf{u} - \eta \nabla_{\mathbf{u}} L(\mathbf{u})$$

Update rule for a single weight for the hidden to output layer

$$\frac{\partial L}{\partial u_{ij}} = \frac{\partial L}{\partial y_j}\,\frac{\partial y_j}{\partial a_j^{(2)}}\,\frac{\partial a_j^{(2)}}{\partial u_{ij}}$$

$$= (y_j - t_j)\, y_j\,(1 - y_j)\, h_i = \delta_j h_i$$

$$u_{ij} \leftarrow u_{ij} - \eta\, \delta_j\, h_i$$



Update rule for a single weight for the input to hidden layer

$$\frac{\partial L}{\partial w_{ij}} = \delta_j\, x_i \qquad \delta_j = \sum_k \delta_k\, w_{jk}\, y_j\,(1 - y_j)\, x_i$$

Errors for hidden to output layer

Errors for input to hidden layer

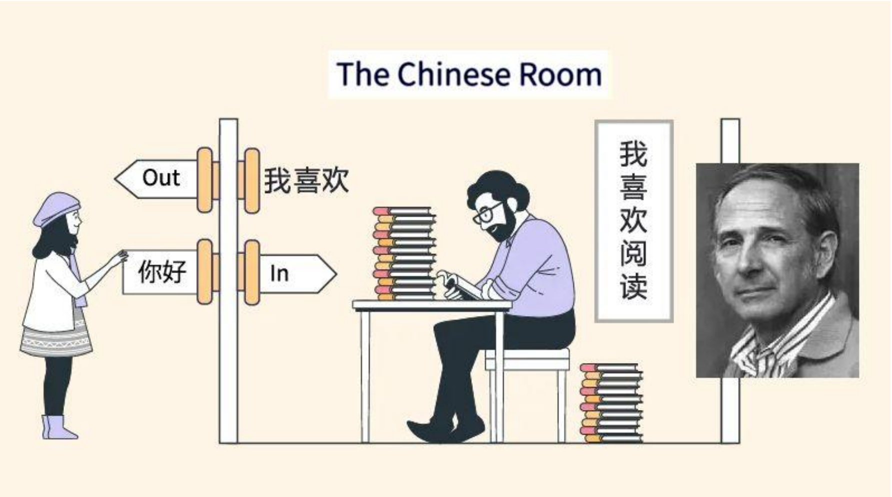
Errors are propagated backward through the network by calculating the gradient of the loss function with respect to each weight using the chain rule of calculus.

First AI Winter (1974–1980) - Challenges

- **Early 1970s**: **Challenges and Critiques**
  - AI programs of the early 1970s struggled to handle complex, real-world problems. They were mostly confined to solving puzzles, playing simple games, or operating in highly constrained environments
  - Insufficient memory and processing power limited scalability (NLP used only 20 words)
  - Computational resources required by symbolic AI algorithms increased exponentially with the size of the problem
  - AI required vast amounts of real-world data, unattainable with 1970s technology
- **1973**: **Funding Cuts**
  - The British government commissioned a report by Sir James Lighthill to evaluate the progress of AI research: Report criticized AI's lack of progress, leading to funding reductions.
  - DARPA Cuts: Disappointed by slow progress, ended major grants like the Speech Understanding Research program.
- **1980** - John Searle presented the **Chinese Room argument**, questioning AI's ability to "understand" symbols.

First AI Winter (1974–1980) – Chinese Room Argument

- **1980** - John Searle presented the Chinese Room argument, a powerful critique against the idea that computers can truly "understand" in the same way humans do
- **The Setup**: Imagine a person who doesn't understand Chinese is locked in a room. This room contains:
    - A large collection of Chinese symbols.
    - A detailed rule book (**in English**) that explains how to manipulate these symbols. The rules specify which symbols to give as output in response to certain input symbols, without requiring any understanding of their meaning.
    - Slots to receive input symbols and provide output symbols to the outside world.



- ***Syntax vs. Semantics***: Computers excel at syntax but lack the semantic understanding required for true cognition.
- ***Strong AI vs. Weak AI***: Challenges "strong AI" (computers possessing true understanding). Supports "weak AI" (computers as tools simulating cognitive abilities).
- ***Mind-Body Problem***: Argues understanding may require more than computation—possibly a biological basis like human consciousness.

Few concepts to learn

Machine Learning Paradigms

Supervised

Learning using labeled data, where each input has a corresponding output. Goal is to learn their relationship and generalize to unseen data

Examples: Classifying the images, Stock market prediction, Medical Diagnosis, Sentiment Analysis

Unsupervised

Learning using unlabeled data. The goal is to find hidden patterns, structures and groupings in the data

Examples: Grouping customers based on purchasing behaviors, Reducing high-dimensional data for visualization

Reinforcement learning

Doesn’t learn from labeled or unlabeled data. An agent learns to make decisions by interacting with environment to maximize a cumulative reward.

Examples: Alpha Go, Chess and Atari Games. Self-driving cars optimizing driving decisions.