

# History and Biological Inspiration

## Neural Networks for Finance

Neural Networks for Finance

BSc Lecture Series

November 30, 2025

## How Does a Committee Make Decisions?

Imagine an investment committee evaluating a stock:

- **Analyst A:** "Strong earnings growth" (+1 vote)
- **Analyst B:** "High debt levels" (-1 vote)
- **Analyst C:** "Good momentum" (+1 vote)
- **Senior Partner:** "Market risk is elevated" (-2 votes)

## The Decision Process:

1. Gather evidence from each analyst
2. Weight opinions by seniority/expertise
3. Sum the weighted votes
4. If total > threshold: **Buy**

## Weighted Voting

Analyst	Vote	Weight
Analyst A	+1	1.0
Analyst B	-1	1.0
Analyst C	+1	1.0
Senior Partner	-1	2.0
<b>Weighted Sum</b>	<b>-1.0</b>	

**Decision: Don't Buy**

---

**Finance Hook:** This is exactly how a perceptron works!

# What If Machines Could Decide?

## The Central Question

In 1943, scientists asked:

*"Can we build a machine that learns to make decisions like a brain?"*

## Why This Matters for Finance:

- Humans are slow and biased
- Markets process millions of data points
- Pattern recognition at scale
- Consistent, emotionless decisions

## The Promise

If we could capture how neurons compute:

- Automatic stock screening
- Risk assessment at scale
- Pattern detection in market data
- Learning from historical decisions

## The Challenge

How do we translate biological processes into mathematical operations?

*This module tells the story of how scientists attempted this translation.*

---

The fundamental question that started neural network research

## The Complete Journey (4 Modules)

### 1. The Perceptron (Today)

- Single neuron foundations
- 1943-1969 history

### 2. Multi-Layer Perceptrons

- Stacking layers, activation functions

### 3. Training Neural Networks

- Backpropagation, optimization

### 4. Applications in Finance

- Stock prediction case study

## Today's Module Structure

### 1. Historical Context (1943-1969)

- McCulloch-Pitts, Hebb, Rosenblatt

### 2. Biological Inspiration

- From neurons to mathematics

### 3. The Perceptron

- Intuition, then math

### 4. Learning Algorithm

- How it adjusts weights

### 5. Limitations

- XOR problem, AI Winter

---

Your journey through neural network fundamentals

By the end of this module, you will be able to:

## 1. Understand biological inspiration

- How real neurons inspired artificial ones
- What we kept and what we simplified

## 2. Master the perceptron model

- Inputs, weights, sum, activation
- The decision-making unit

## 3. Interpret decision boundaries

- Geometric meaning of weights
- Linear separability concept

## 4. Apply the learning algorithm

- Weight update rule
- Convergence conditions

## 5. Recognize limitations

- XOR problem
- Why single layers are not enough

**Finance Connection:** Throughout, we'll use stock classification as our running example.

---

By the end of this module, you will be able to...

# 1943: The Mathematical Neuron

## Warren McCulloch & Walter Pitts

In 1943, a neurophysiologist and a logician asked:

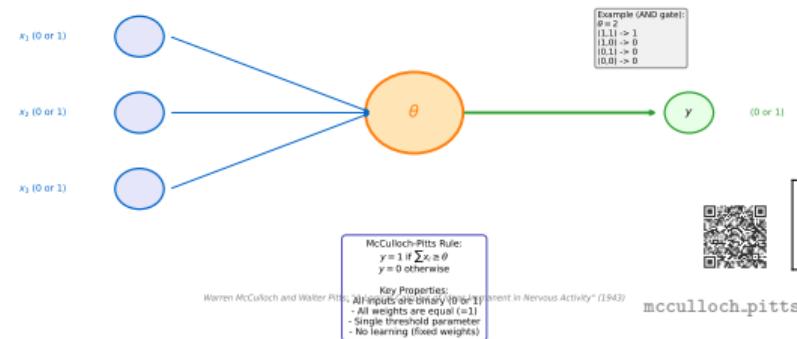
*“Can we describe what neurons do using mathematics?”*

Their paper: “A Logical Calculus of Ideas Immanent in Nervous Activity”

### Key Insight:

- Neurons have binary states (fire or not)
- This is like TRUE/FALSE in logic
- Networks of neurons can compute any logical function

McCulloch-Pitts Neuron (1943): Binary Threshold Logic



.../.

---

Warren McCulloch and Walter Pitts: “A Logical Calculus of Ideas Immanent in Nervous Activity”

## What McCulloch & Pitts Proposed

The brain performs computation through:

### 1. Binary Signals

- Neurons either fire (1) or don't (0)
- Like bits in a computer

### 2. Threshold Logic

- Sum of inputs exceeds threshold  $\rightarrow$  fire
- Otherwise  $\rightarrow$  stay quiet

### 3. Network Composition

- Complex behaviors from simple units
- AND, OR, NOT gates from neurons

## Logical Operations with Neurons

**AND Gate** (threshold = 2):

- Both inputs = 1  $\rightarrow$  output = 1
- Otherwise  $\rightarrow$  output = 0

**OR Gate** (threshold = 1):

- Any input = 1  $\rightarrow$  output = 1
- All inputs = 0  $\rightarrow$  output = 0

**Implication:** If neurons compute logic, and computers compute logic, then we can build artificial brains!

---

If neurons compute, can we build artificial ones?

# 1949: Hebbian Learning

## Donald Hebb's Insight

McCulloch-Pitts neurons were fixed. But how does the brain *learn*?

### Hebb's Rule (1949):

*"Neurons that fire together, wire together."*

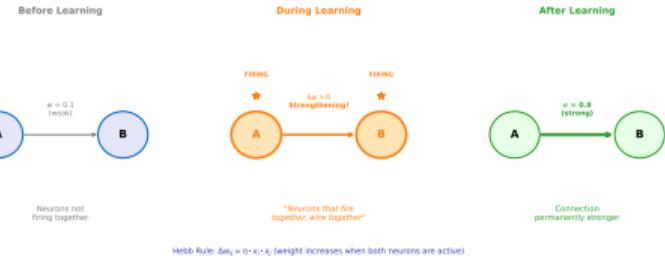
### In Plain Terms:

- If neuron A consistently activates neuron B
- The connection  $A \rightarrow B$  grows stronger
- Repeated patterns reinforce pathways

### Finance Analogy:

An analyst who repeatedly identifies winning stocks gains more influence in the committee.

Hebbian Learning: "Neurons That Fire Together, Wire Together" (Donald Hebb, 1949)



.../.

[hebb.learning-visualization.com](http://hebb.learning-visualization.com)

Donald Hebb: "Neurons that fire together, wire together"

# 1958: The Perceptron is Born

## Frank Rosenblatt at Cornell

Combined McCulloch-Pitts neurons with Hebbian learning into a machine that could *learn from examples*.

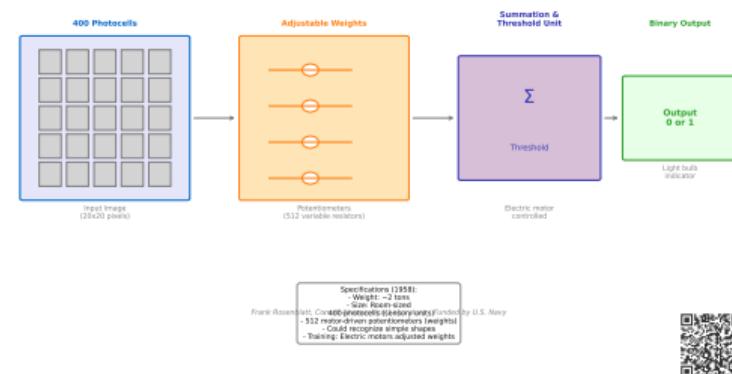
### The Perceptron:

- A single artificial neuron
- Adjustable connection weights
- Learns to classify patterns
- Implemented in hardware (Mark I)

### Key Innovation:

Not just fixed logic gates, but a system that **learns** the right weights from training data.

Mark I Perceptron (1958): First Neural Network Hardware



...

The Mark I Perceptron used 400 photocells connected to a single layer of neurons with adjustable weights.

Frank Rosenblatt creates a machine that can learn

# The New York Times Headline

July 8, 1958 - The New York Times

*"New Navy Device Learns By Doing; Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser"*

## The Promises Made:

- Machines that recognize faces
- Automatic translation of languages
- Systems that “perceive” like humans
- The Navy predicted: walking, talking, self-reproducing machines

## The Reality:

The perceptron could classify simple patterns, but the gap between promise and capability was vast.

## Lessons for Today

### Sound Familiar?

- “AI will replace all jobs”
- “Machines will be smarter than humans by 20XX”
- “This changes everything”

### Pattern:

1. Genuine breakthrough
2. Media amplification
3. Overpromising
4. Disappointment
5. “AI Winter”

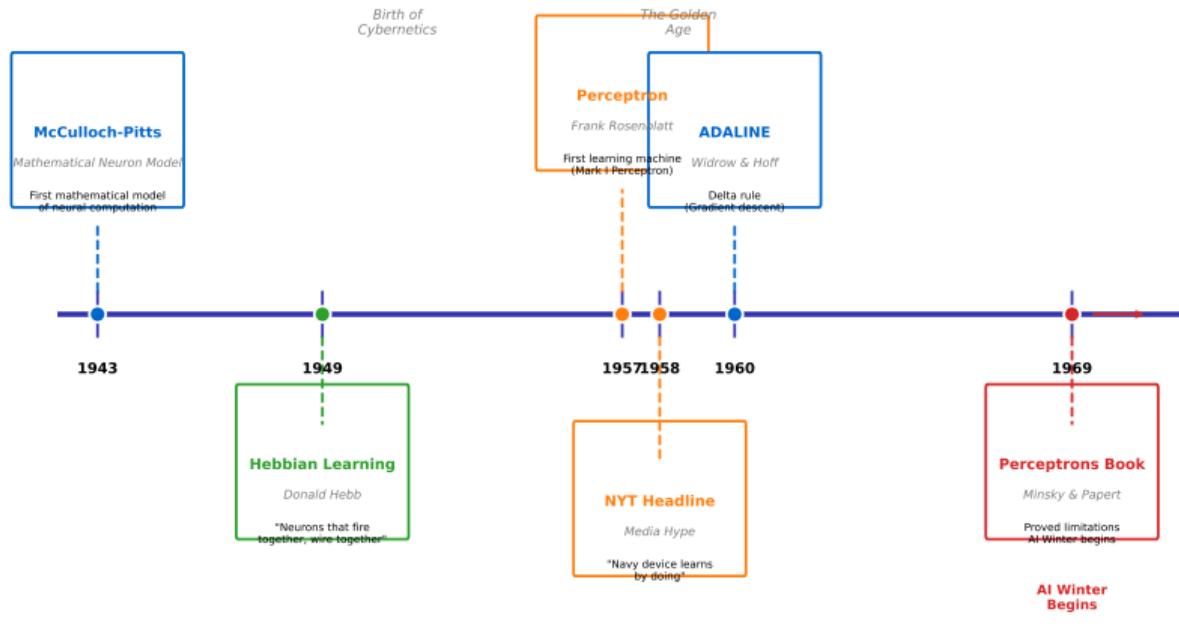
*History repeats...*

---

“New Navy Device Learns By Doing” - The hype cycle begins

# Timeline: The Early Years

## Neural Networks: The Early Years (1943-1969)



...

timeline.1943-1969

From theory to hardware in 15 years

*"The perceptron was funded by the US Navy for military applications. How does funding source shape research direction? Are there parallels in modern AI development?"*

Consider:

- Military vs. commercial vs. academic funding
- What problems get prioritized?
- Open vs. closed research
- Today: Tech giants fund most AI research
- Government initiatives (CHIPS Act, etc.)
- Startup ecosystem influence

---

Think-Pair-Share: 3 minutes

## Anatomy of a Real Neuron

### 1. Dendrites (Input)

- Tree-like branches
- Receive signals from other neurons
- Thousands of connections

### 2. Cell Body (Soma) (Processing)

- Integrates incoming signals
- Contains the nucleus
- Determines if neuron fires

### 3. Axon (Output)

- Long fiber carrying output signal
- Connects to other neurons
- All-or-nothing signal

## How It Works

1. Signals arrive at dendrites
2. Soma sums the inputs
3. If sum exceeds threshold: neuron **fires**
4. Action potential travels down axon
5. Signal reaches next neurons

## Key Numbers:

- Human brain: ~86 billion neurons
- Each neuron: ~7,000 connections
- Total synapses: ~100 trillion

---

Dendrites receive, soma processes, axon transmits

## Mathematical Abstraction

### 1. Inputs ( $x_1, x_2, \dots, x_n$ )

- Numerical values (features)
- Replace dendrites

### 2. Weights ( $w_1, w_2, \dots, w_n$ )

- Importance of each input
- Replace synapse strength

### 3. Weighted Sum

- $z = \sum_{i=1}^n w_i x_i + b$
- Replace soma integration

### 4. Activation Function

- $y = f(z)$
- Replace firing decision

## The Complete Model

$$y = f\left(\sum_{i=1}^n w_i x_i + b\right)$$

### Components:

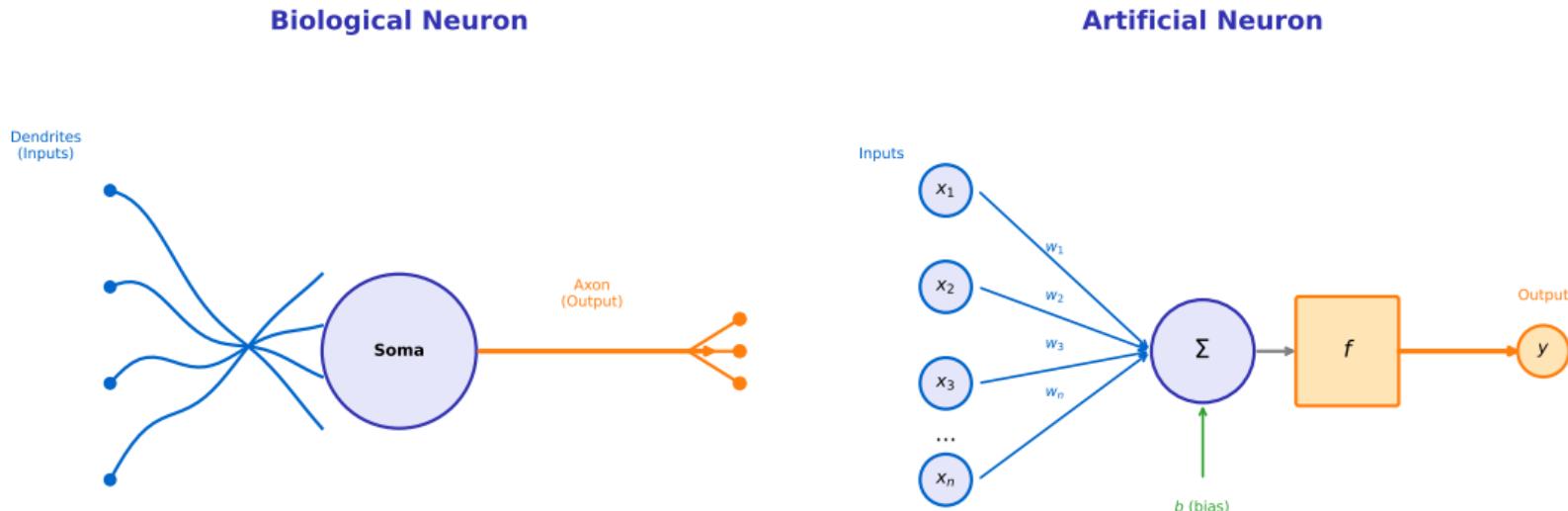
- $x_i$ : Input features
- $w_i$ : Learnable weights
- $b$ : Bias (threshold adjustment)
- $f$ : Activation function
- $y$ : Output (prediction)

**Key Point:** The weights are what the network *learns*.

---

From biology to mathematics: the abstraction trade-off

# Biological vs. Artificial: Side by Side



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



...

[biological\\_vs\\_artificial\\_neuron.pdf](https://biological_vs_artificial_neuron.pdf)

What did we keep? What did we simplify?

## A Financial Analyst as a Neuron

Biology	Finance
Dendrites	Market data feeds
Synapses	Data reliability weights
Soma	Analyst's judgment
Threshold	Conviction level
Axon	"Buy" recommendation

### The Process:

1. Receive multiple data points
2. Weight by source quality
3. Aggregate into overall view
4. If conviction > threshold: recommend

## Example: Stock Screening

### Inputs (Data):

- $x_1$ : P/E ratio = 15
- $x_2$ : Revenue growth = 20%
- $x_3$ : Debt/Equity = 0.5

### Weights (Importance):

- $w_1 = 0.3$  (value focus)
- $w_2 = 0.5$  (growth priority)
- $w_3 = -0.2$  (debt penalty)

### Decision:

$$z = 0.3(15) + 0.5(20) - 0.2(0.5) = 14.4$$

If  $z > 10$ : **Buy**

---

Inputs (data) -  $\downarrow$  Weights (importance) -  $\downarrow$  Decision (output)

## Benefits of Simplification

### 1. Mathematical Tractability

- We can write equations
- Analyze behavior formally
- Prove theorems

### 2. Computability

- Easy to implement in code
- Fast computation
- Scales to millions of units

### 3. Trainability

- Can adjust weights systematically
- Gradient-based optimization
- Learn from data

## What We Can Now Do

- Define learning algorithms
- Compute exact outputs
- Train on historical data
- Make predictions on new data
- Analyze decision boundaries

## Scale Comparison:

	Brain	GPU
Operations/sec	$10^{16}$	$10^{15}$
Power	20W	300W
Training time	Years	Hours

Different trade-offs, different capabilities.

---

Simplification enables computation

## Biological Complexity We Ignored

### 1. Temporal Dynamics

- Real neurons have timing
- Spike patterns carry information
- We use static activations

### 2. Structural Complexity

- Dendrites have local computation
- Different neuron types
- We use uniform units

### 3. Neurochemistry

- Neurotransmitters vary
- Modulatory systems
- We use simple multiplication

## Implications

### What ANNs Cannot Do (Well):

- Energy efficiency of brain
- One-shot learning
- Continuous adaptation
- Common sense reasoning

### The Trade-off:



*Artificial neurons are inspired by biology, not copies of it.*

---

The brain does far more than our models capture