



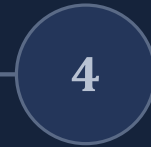
Neurons to Networks



How Networks Learn



Language



Agents

From NN to LLM: Rise of the Machines

Lecture 1: Neurons to Networks

1906 to 1969

AI in Context, Module 2

Vishal Misra, Department of Computer Science, Columbia University

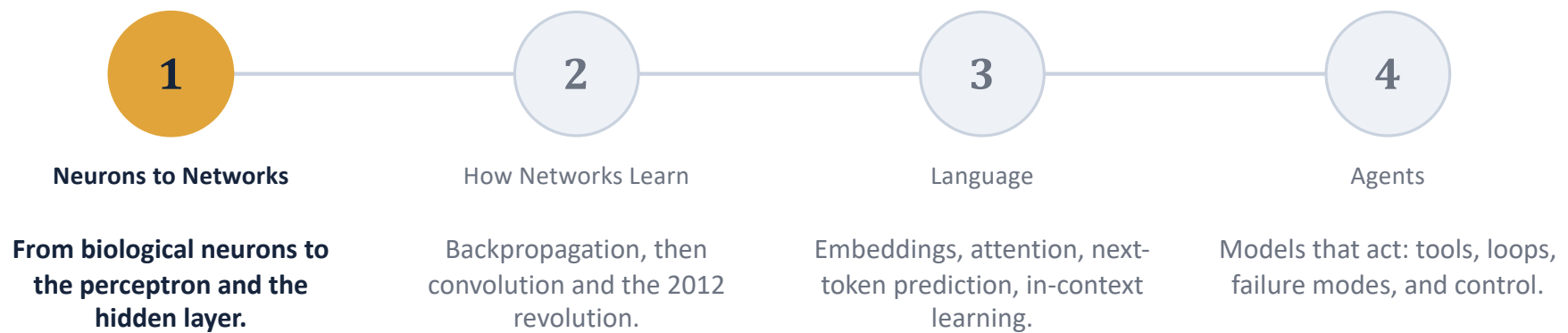
One sentence in, a working program out

```
Build a Jupyter notebook I can open in Colab.  
Train a neural network to recognize handwritten  
digits, report its accuracy, then show me 10  
digits it gets wrong.
```

- It will plan, write code, run it, read the errors, and fix them.
- That is the course project, built in a few minutes.
- By Lecture 4 you can explain every step.

Today, watch what it does. How it does it is the next four lectures.

From neurons to agents

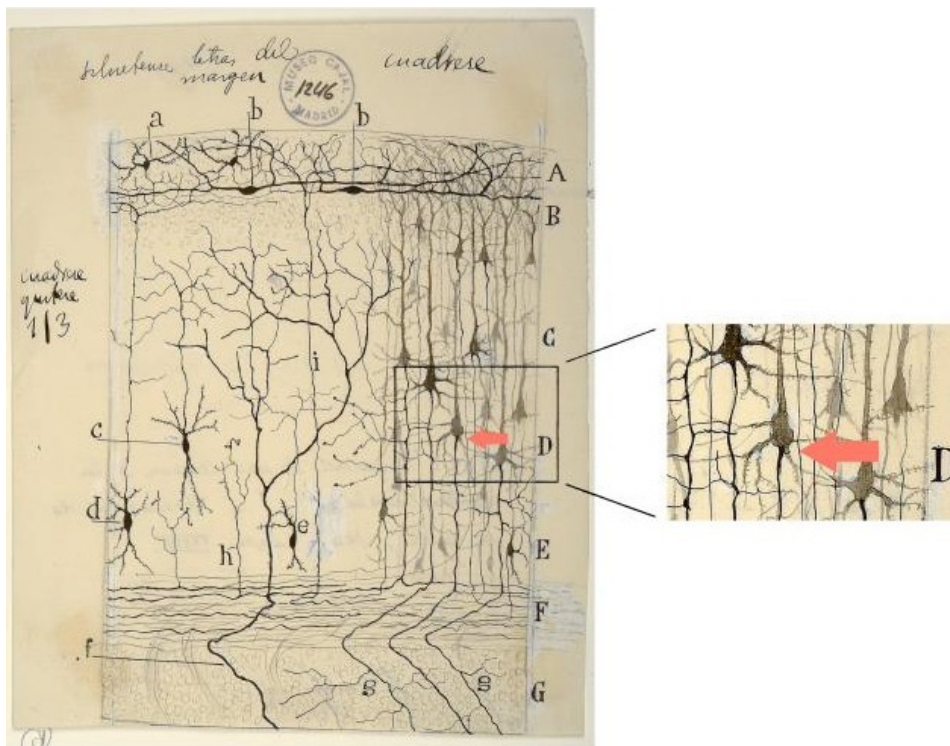


1906

The brain is made of separate cells

Santiago Ramón y Cajal wins the Nobel Prize for showing it.

Cajal: neurons are separate cells



- Drawn by hand from a microscope, cortex tissue stained with silver.
- Each neuron is a separate cell.
- Cells talk across gaps, later named synapses.
- Shared the 1906 Nobel Prize with Camillo Golgi, who disagreed.

A biological neuron

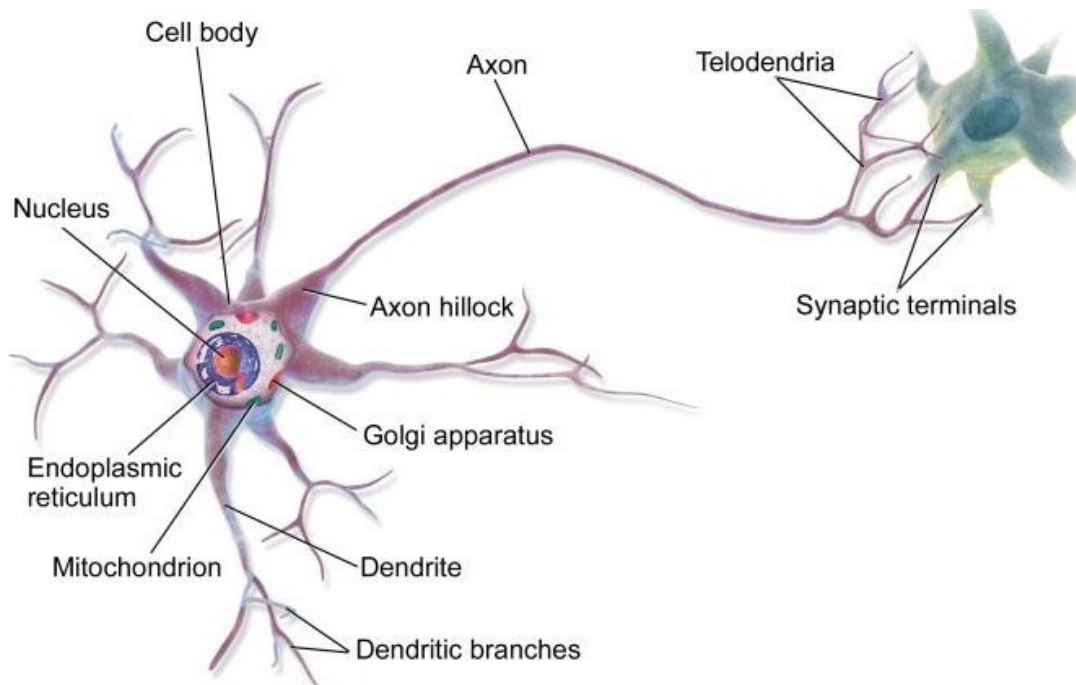


Image: BruceBlaus, CC BY 3.0, Wikimedia Commons

86 billion

neurons in a human brain

$\sim 10^{14}$

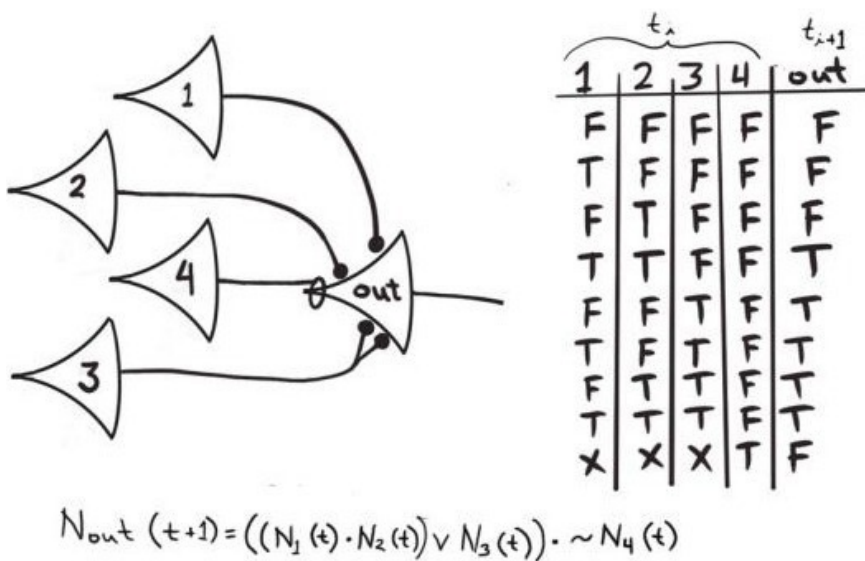
synapses, the connections between them

1943

A neuron as a logic gate

Warren McCulloch, a neurophysiologist, and Walter Pitts, a self-taught logician.

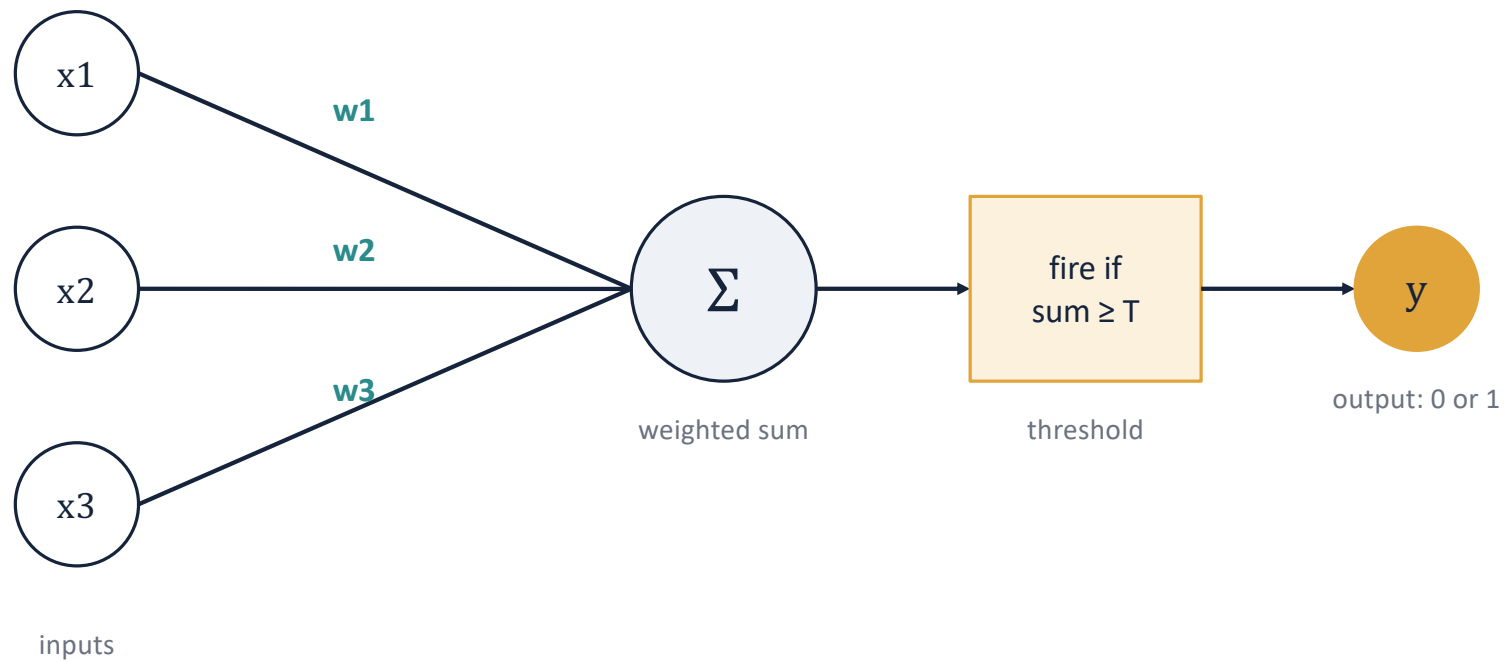
McCulloch and Pitts: a logical calculus of neurons



- Each neuron is on or off.
- It fires when enough inputs fire.
- Networks of them can compute any logical expression.
- The weights are set by hand. No learning yet.

Figure from McCulloch and Pitts (1943): a small network and its truth table

The artificial neuron



$$y = 1 \text{ if } w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 \geq T, \text{ else } 0$$

Choose the weights, get a logic gate

Gate	Weights	Threshold T	Fires when
AND	$w_1 = 1, w_2 = 1$	2	both inputs are on
OR	$w_1 = 1, w_2 = 1$	1	at least one input is on
NOT	$w_1 = -1$	0	the input is off
XOR	?	?	exactly one input is on

Change the weights and the same unit computes a different function. The weights are the program.

1949: Hebb proposes how the brain learns

“Neurons that fire together, wire together.”

Hebb's actual wording was longer

In the brain

When one cell repeatedly helps fire another, the connection between them strengthens.

In the model

Learning is a change in weights. Memory is in the connections.

1958

The perceptron learns its own weights

Frank Rosenblatt, Cornell Aeronautical Laboratory, funded by the US Navy.

The perceptron learning rule

```
error = target - output  
weight ← weight + α × input × error
```

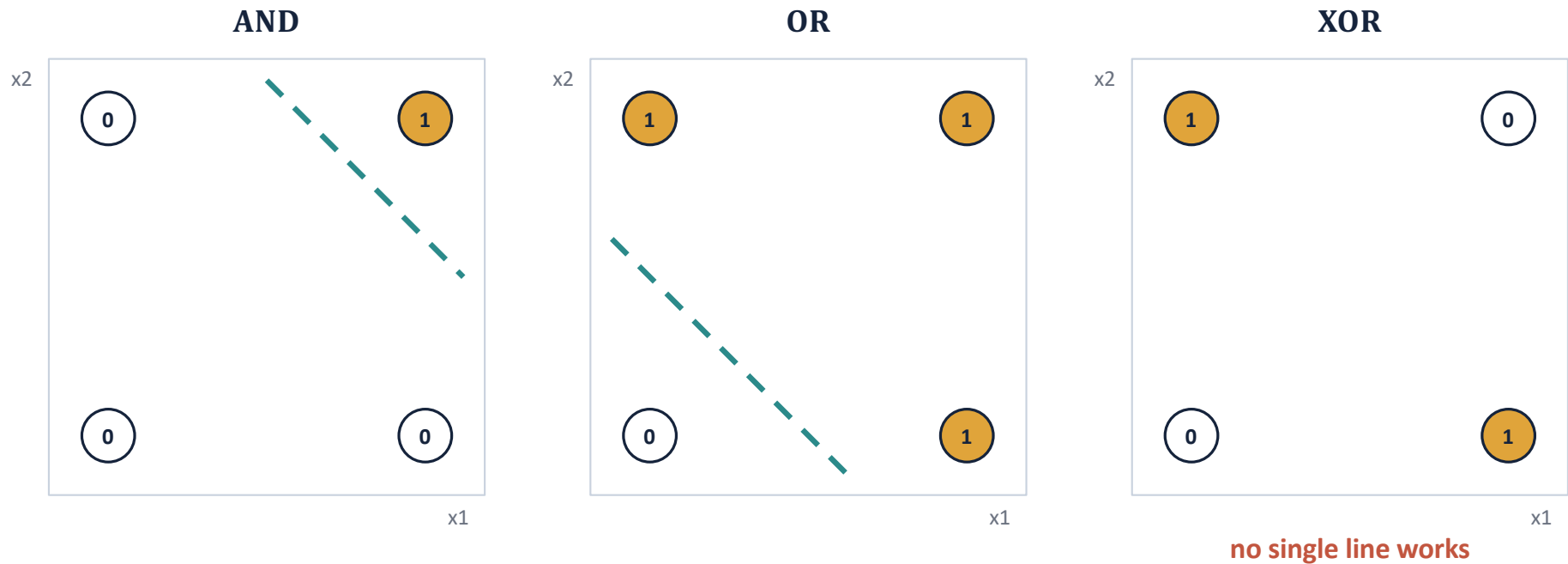
- Guess. Compare with the right answer. Nudge the weights toward it.
- No one sets the weights by hand anymore.
- Built in hardware: the Mark I Perceptron read a 20×20 grid of photocells, 400 inputs.

The New York Times, July 1958

The Navy expected the machine to “walk, talk, see, write, reproduce itself and be conscious of its existence.”

It could tell left from right.

A perceptron draws a straight line

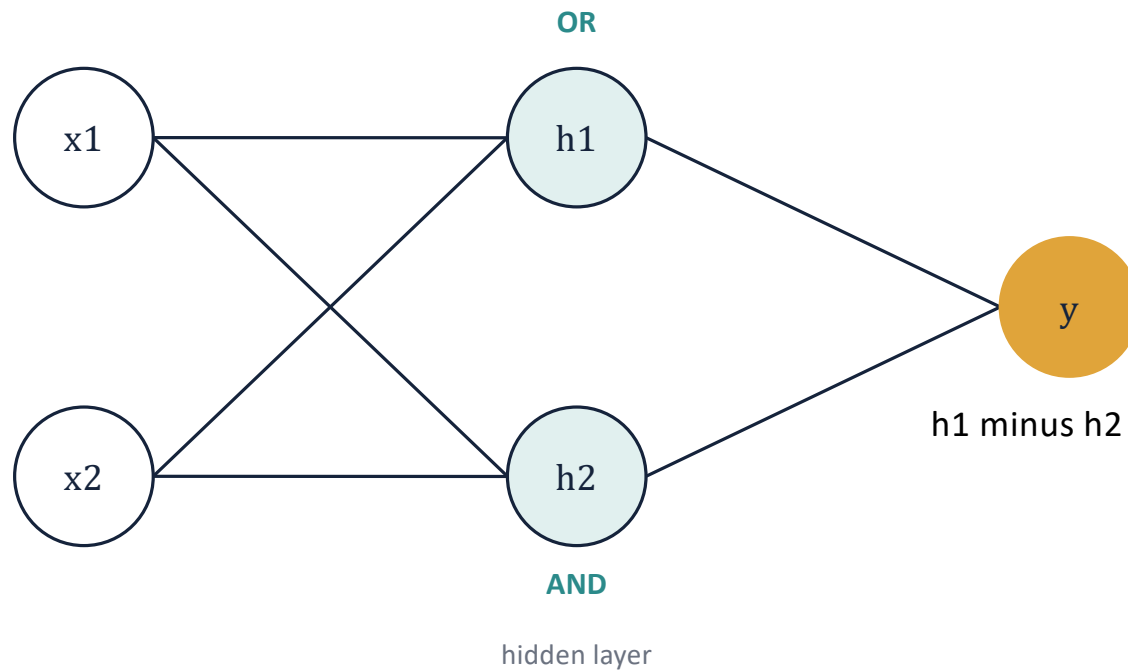


$w_1 \cdot x_1 + w_2 \cdot x_2 = T$ is a line. One side fires, the other does not.

1969: Minsky and Papert prove a single-layer perceptron cannot compute XOR.

Funding for neural network research falls away through the 1970s. The fix was already known in principle: add a layer. Nobody knew how to train one.

The fix: add a hidden layer



x1	x2	h1	h2	y
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

The hidden layer solves XOR by breaking it into simpler pieces.

Same name, different machine

A real neuron

- Spikes: timed pulses rather than numbers.
- Modeling one cell takes a 5 to 8 layer network (Beniaguev et al., 2021).
- Its dendrites alone can compute XOR (Gidon et al., 2020).
- Usually excitatory or inhibitory, with known exceptions.
- The whole brain runs on about 20 watts.

An artificial neuron

A weighted sum and a threshold.
Needs a second layer to compute XOR.
Weights can be positive or negative.
The 1943 name stuck.

Inspiration is not identity.

Your project: teach a network to read digits

70,000

handwritten digits in MNIST: 60,000 to train,
10,000 to test

784

inputs per image: 28×28 pixels

10

outputs: one per digit, 0 to 9

- Work in the Colab notebook linked on the course site. It runs in the browser.
- Part 1 this week: a single-layer network. Record its accuracy.
- Part 2 next week: add hidden layers, then convolution. Compare.

Next: a network with a hidden layer can compute XOR. How does it learn the weights?

Lecture 2: backpropagation, then the images that launched deep learning.