˚ ₊‧ৎ❀ A Cozy Walk Through the History of AI ♡

Divya karlapudi 10 min read · Nov 22, 2025

A journey from neurons to transformers ❀

⋆˚🌀｡ **Introduction:**

When we think of AI today, we picture ChatGPT answering questions like a friend, self-driving cars navigating traffic, and deepfake videos that looks scarily real. But AI didn't just appear one morning. It wasn't magic. It was built, one idea at a time.

For decades, researchers often working quietly in labs kept pushing the boundaries of imagination and computation. **And yet, when we try to learn AI today, it's easy to feel overwhelmed**. There are too many buzzwords, too many frameworks, too many "must-learn" topics.

So… what if we **start from the beginning** instead? ᵔᵕᵔ

⤷ Where did these ideas come from?

⤷ Why were they created?

⤷ What problems were researchers trying to solve?

Understanding the journey, not just the destination makes AI surprisingly human, relatable, and even comforting. (**yes, comforting** (´｡• ◡ •｡`) ♡)

❄️ **If we first understand the problems that existed.. the solutions begin to make sense.**

So, let’s start with history. **The ideas, the inspirations, the breakthroughs.** Let’s understand how AI slowly went from theory… to code… to reality 🥹

Ready? ٩(ˊᗜˋ*)و ♡

⚡ **It was 1943.. the first artificial neuron was born.**

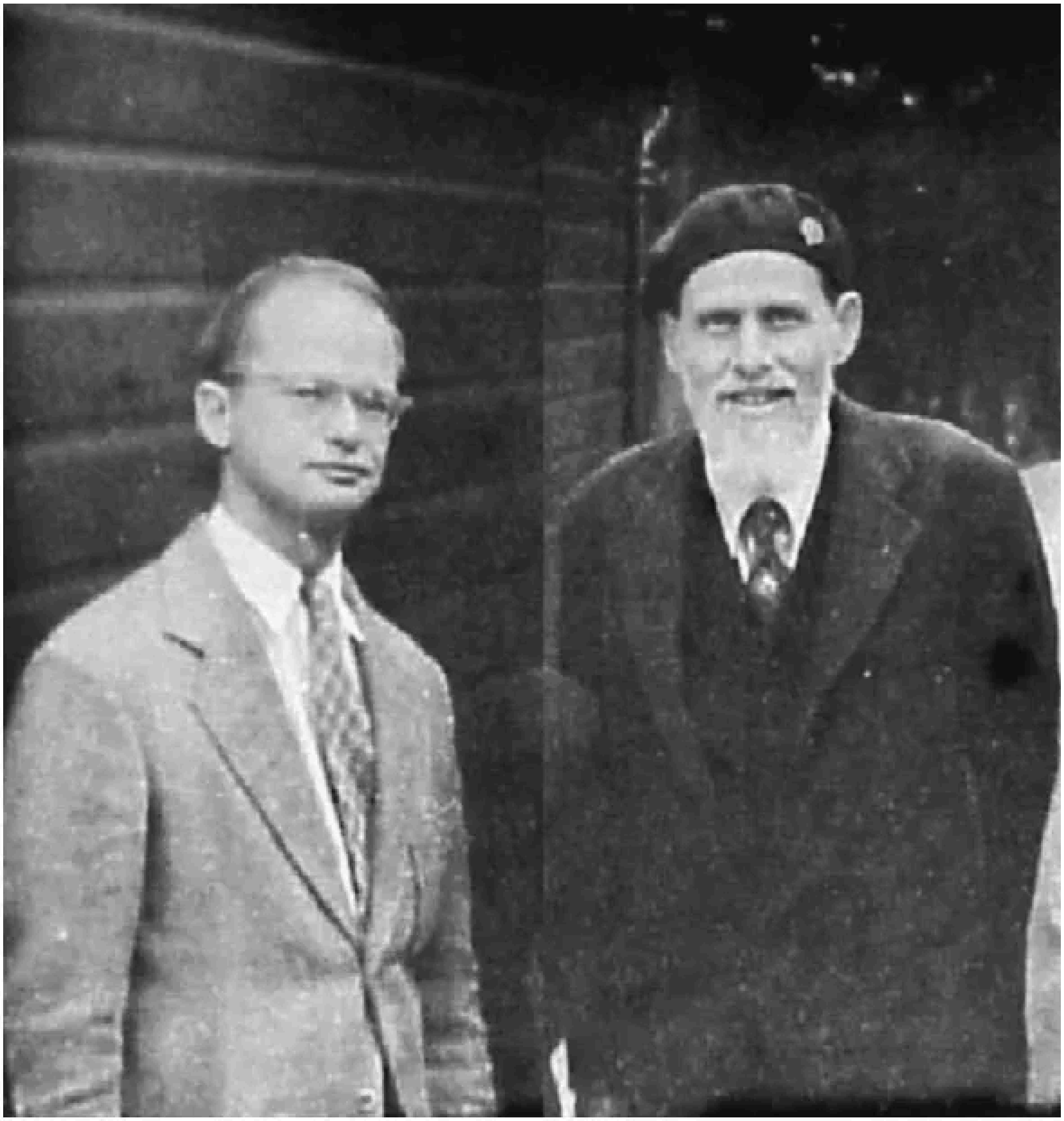
Sounds unbelievable, right?

1943. No internet. No GPUs. No Python.

Yet… the first concept of an artifical neuron was already on paper 🌟

That year, **Warren McCulloch** (a neurophysiologist) and **Walter Pitts** (a young mathematician with no formal degree) wrote a groundbreaking paper titled:

“A Logical Calculus of the Ideas Immanent in Nervous Activity”



Walter Pitts (left) & McCulloh (Right)

This was the birth of the first mathematical model of a neuron (**McCulloch-Pitts neuron**).

It didn't learn. It didn't adapt. But it was logical (**literally**). It used **AND, OR, NOT** operations just like today's digital circuits.

Why was this a big deal (🤔)?

Because it showed something revolutionary:

(🧠 Maybe the brain could be explained using logic.
And if the brain is logical… maybe a machine could think too.)

This was the first time anyone treated **thinking** as something mathematically computable.

👉 *This paper lit the first spark of Artificial Intelligence.*

Here's the actual paper ʿଓ : A Logical Calculus of the Ideas Immanent in Nervous Activity (1943)

From this humble beginning… the road to AI officially started to unfold 🪄

———୨୧———

✨ **1958. The perceptron brings hope** ✿

It's the late 1950s. **Computers are huge**, **expensive**, and **painfully slow**.

AI isn't even a defined term yet.

But then, a young **psychologist** at Cornell named **Frank Rosenblatt** introduces something bold… (He was originally a psychologist, not a computer scientist >ᴗ<)

A machine that could learn from data. A machine inspired by how the human brain works.

A machine called “**The perceptron**” 🧨

🎉 Why Was This a Big Deal?

For the first time in history, a system could learn patterns from inputs and make decisions.

🌸 The perceptron was basically an early **single-layer neural network** and, the idea was beautiful:

It could **recognize simple shapes**, **classify patterns**, and **adjust its weights** over time. Which means… **It could learn.** (That word “learn” changed everything.)

What Could the Perceptron Do? 🤔

Let’s say we show it multiple samples of **circles** and **triangles.**

Over time, it learns to say: “**Yes, this looks like a circle**” Or, “**Nope.. that’s more like a triangle.**”

Sounds basic today. But in 1958, this felt like magic 🥹

This was the first step towards modern neural networks.

Frank Rosenblatt

The perceptron was powerful, but there was a problem ☝️ :

It couldn’t learn complex patterns like XOR functions.

Here’s the original paper ᥫ᭡ : “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain”

Right now, in **1958, AI was full of hope.** Our, Next Stop **1960s & 1970s: The First AI Winter** 🧊

———୨ৎ———

Let’s slow down the timeline and **feel what happened in 1969**, because this moment **didn’t just slow AI down… It froze it** 🍂

❄️ 1969. The First AI Winter Begins…

The excitement from **Rosenblatt**’s perceptron was still echoing across research labs.

Many believed machines would soon **think**, **learn**, even **see like humans**.

But then…

Two researchers “**Marvin Minsky**” and “**Seymour Papert**” published a book that would reshape AI history.

📕 **“Perceptrons” (1969):** A mathematical and critical analysis of neural networks.

Minsky & **Papert** mathematically proved something devastating:

🧠 **Single-layer perceptrons cannot learn certain simple functions** (like XOR).

That means:

↳ They could only separate things **linearly.**

↳ If data was **complex** or **overlapping?** → **It failed**

↳ It couldn’t understand relations like:
“If both inputs are different, output should be 1.” (XOR)

In short: **The perceptron was more limited than everyone thought** (·•́_•̀)

Consequences?₊🥀

Government funding agencies lost hope. Investors pulled back. Universities stopped encouraging AI research.

This period is known as **the First AI Winter (1969–1980s)** because, **interest froze. Funding froze. Careers froze.**

Many researchers moved on to other fields like **Statistics**, **Expert systems…** **Neural networks** were abandoned 😟

.

Marvin Minsky (left) and Seymour Papert (right)

Their critique was correct. But, only for single-layer neural networks. They didn't consider multi-layer networks (MLPs) Or, backpropagation because, these concepts weren't even discovered yet!

So.. AI wasn’t dead. It was simply waiting… for the next breakthrough 🩷

Here’s the **original scan of the book** ᡣ𐭩 **:** **Perceptrons — An Introduction to Computational Geometry**

———୨ৎ———

✈️ **1980s. AI winter is about to melt…** ε(´｡•ᴗ•`)っ 🎀

After the criticism of perceptrons in 1969, many believed neural networks were useless.

But in **Japan**, one researcher wasn’t ready to give up ✨

Kunihiko Fukushima. The Quiet Revolutionary (personal fav) ᐠ(｡> ᵕ <)ᐟ♡

.

Kunihiko Fukushima

In 1980, **Kunihiko Fukushima** introduced something groundbreaking.

🧠 **The Neocognitron.** (A multi-layered neural architecture deeply inspired by **the human visual system**).

This architecture became the ancestor of modern **Convolutional Neural Networks** (CNNs) ♡

What Was the Neocognitron? 🤔

It was designed to recognize patterns like shapes and handwritten characters **even if they were shifted**, **distorted**, or **scaled** (well.. to a large extent).

It consisted of layers 🗐 (where **S-cells** (simple cells), detect basic features like edges.. **C-cells** (complex cells), become invariant to position and shape.. **Hierarchical structure**.. Increasing abstraction across layers)

(No.. no.. don’t get overwhelmed, you will get it if you read the paper)

👉 This structure is very similar to modern **CNNs** like **LeNet-5** & **ResNet**!

Fig explaining the interconnections between layers. (from Neocognitron paper)

Here’s the paper ⁏ঌ : “**Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position**”

Even though the **Neocognitron** was brilliant, It **learned via local rules but wasn’t fully efficient or scalable.**

We were adjusting weights with **no systematic method to minimize error.**

🫧 **Imagine:** We feed an image of a “**cat**” into our neural network, if it outputs “**dog**” i.e., total error is high but, the question is **Which weight caused this mistake? Which layer is at fault? How much should each weight change?**

Without an answer to these questions, learning was impossible. Especially in deep layers.

We needed a method that could **compute the error, trace it back through all the layers** and **adjust each weight precisely.**

Let’s step into **1986, a year that revived AI.** It was like the moment someone found the missing key… and **opened the door again** 🚪 ✨

———୨୧———

🚀 1986. Backpropagation Brings Neural Networks Back to Life

🧑‍🔬 Three researchers “**Geoffrey Hinton**”, “**David Rumelhart**”, and “**Ronald Williams**” have officially introduced **backpropagation** as a practical training method for multi-layer neural networks.

How exactly backpropagation works?

1. Feed input forward (get prediction)
2. **Compare with the correct answer** (calculate error)
3. **Send the error backwards through layers**
4. **Adjust each weight using calculus**

5. Repeat until the error is small.

This allowed networks to actually learn, rather than guess.

What did they show? ⚡

They demonstrated that Multi-layer neural networks **can learn complex patterns, Deep layers can be trained using gradient descent, Errors can be propagated backward to adjust every weight** 🏆

`This paper is considered **a turning point in AI history**`

Here’s the paper ‘ଓ : “**Learning representations by back-propagating errors**”

———୨୧———

“**But, can a network remember?** Can it handle sequences, language, time?” 🤔

🔮 **“What if neural networks could not only think… but also remember?”**

Backpropagation (**1986**) changed everything. It allowed neural networks to finally **learn, layer by layer**.

But soon, researchers hit a new limitation: **Standard neural nets looked at data as if every input was independent.**

↳ A word has no connection to the previous word.

↳ A video frame knew nothing about the last frame.

↳ A time series point didn’t know it’s past.

That’s not how we humans think.. We understand sequences, context, stories, time.

So researchers asked: “**Can we build a neural network that learns over time?**”

🌸 **Recurrent Neural Networks (RNNs). First Step Towards Intelligence.**

Between **1986** and **1990**, **Recurrent Neural Networks (RNNs)** were introduced. Two major names appear here.

“**Michael I. Jordan**” **: 1986 :** Introduced the **Jordan Network** (a neural network that loops it’s output back as input)

Michael I. Jordan

“Jeffrey L. Elman” : 1990 : Introduced the **Elman network** (used a hidden “**context layer**” to preserve memory.

Jeffrey L. Elman

They both had the same core philosophy: **“Let the network remember it’s previous state”**

This wasn’t just brilliant. **It was the first computational model of memory in AI** 👏

But, how RNNs work? 😳

Imagine reading a sentence. **Word by word.** Our brain doesn’t forget the first word when reading the second one.

RNNs try to replicate this. They take input, **update an internal state**, and pass it forward.

Input → **Hidden State (memory)** → Output → **Next Hidden State** → …

🎉 This little loop became groundbreaking as it allowed AI to **process Speech**, **Text**, **Time-series data**, **Sensor signals**, **Music…**

🚧 **But There Was a Problem…** (you saw it coming right? 😂)

Training **RNN**s with **regular backpropagation** failed badly.

The gradients became:

1. **Too small** → **network forgot everything** (vanishing gradient)
2. **Too large** → **network exploded** (exploding gradient)

The Result? : RNNs could remember.. but only short-term memory.

When they try to remember information across many time steps, they fail.

Let's imagine this sentence:

"The book that the professor who I met yesterday recommended was fascinating."

To understand *"was fascinating,"*
you need to remember **"the book…"** from *way earlier* in the sentence.

But, **RNNs** ***forget too quickly*.** As the sequence grows:

Old information **fades.** Only **recent inputs are remembered.**

Why? Because of **vanishing gradients.**

Here are the paper links ꒰ᐢ :

Michael I. Jordan (1986): "**SERIAL ORDER: A PARALLEL DISTRIBUTED PROCESSING APPROACH**"

Jeffrey L. Elman (1990): "**Finding Structure in Time**"

———୨୧———

Welcome to the **1990s**, the first time **AI finally proved itself in the real world** ⋆✶˚｡⋆

🧠 1997: The Birth of Long Short-Term Memory (LSTMs)

By the **mid 1990s**, researchers had hit a wall. **RNNs** failed to handle **long-term dependencies** because of the **vanishing gradient problem.**

That's when "**Sepp Hochreiter**" & "**Jurgen Schmidhuber**" proposed a breakthrough architecture in **1997.**

Long Short-Term Memory (LSTM): A neural network that learns what to remember… and what to forget. (How cool is that? (〃ᐢ ᵕ ᐢ〃))

This changed AI forever. It enabled deep learning in language, speech, translation, music, handwriting, and later, **paved the way for GPT and Transformers.**

🔍 What Was the Core Idea?

Instead of blindly passing memory from step to step (like RNNs), **LSTMs control memory using gates.**

Just like how our brain **filters important events** and **forgets irrelevant noise.**

1. **Memory Cell:** Long-term memory.
2. **Gates:** Attention / decision making.
3. **Forget Gate:** "This isn't important."
4. **Input Gate:** "Store this!"
5. **Output Gate:** "Use this now."

LSTMs overcame **two major limitations in RNNs. Vanishing gradients** and **rapid memory loss.**

They introduced a new concept: **selective memory**, choosing what to forget and what to store.

Enabled neural networks to **retain information across dozens, even hundreds of time steps.** 👏

Here’s the paper ঌ : “**LONG SHORT-TERM MEMORY**”

———୨୧———

The **1986 backpropagation** breakthrough **revived** neural networks. But, scientists still asked: **“Can it actually solve real problems?”** 🪶

In the **early 1990s**, someone answered that question.

⭐ **Yann LeCun. The Architect of Modern Computer Vision.**

Yann LeCun (Photo by Nick Fetty/The New York Academy of Sciences)

A passionate researcher at **AT&T Bell Labs**, he introduced **a neural network that could read handwritten digits automatically**.

This was **LeNet-5.** One of the first **convolutional neural networks** (CNNs) **used in production.**

What was so special about LeNet-5? 💭

It could read **ZIP codes** from envelopes. **Completely automatically.** Used by the **U.S. Postal Service**, no human intervention needed!

↳ **Convolutional Layers:** Detect patterns like edges & shapes.

↳ **Pooling Layers:** Make model resistant to shifts/distortions.

↳ **Backpropagation:** Enabled training end-to-end

↳ **Real-world data:** First practical AI success

This was a huge break-through °❀⋆.ೃ࿔*:･

Here's the paper ᡣ𐭩 : "**GradientBased Learning Applied to Document Recognition**" (1998)

Also, check out: "**Convolutional Networks and Applications in Vision**" (2023)

From **ImageNet** and **AlexNet** to **GANs**, **ResNet**, **Transformers**, **Retrieval-Augmented Generation**, and eventually **LLM agents**… AI evolved rapidly, step by step.

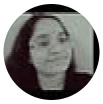
These exciting advancements deserve their own deep dives… and that's exactly what we'll explore in our upcoming articles ❤

I really hope this article gave you a comfortable and confident start on your AI journey ⋆˚✿₊°

With Love ᯓ⁀∘∘∘♡

Divya Karlapudi.

History Of Ai   Artificial Intelligence   Rnn   Cnn   Lstm Model



Written by Divya karlapudi

3 followers · 33 following

Edit profile

I love sharing what's in my mind in general.

No responses yet

Divya karlapudi

What are your thoughts?

More from Divya karlapudi

Divya karlapudi

It’s divinely being alone!

by once a people pleaser.

Oct 29, 2023 👏 2

See all from Divya karlapudi

Recommended from Medium

Myo Oo

E8 Consciousness: I Predicted It. Now I’ve Proved It.

In AI Advances by Nikhil Anand

This embarrassingly simple secret explains all of AI.

How I discovered consciousness is built on E8 geometry

6d ago 👏 51 💬 2

What your CS professor never told you about ChatGPT.

✦ 5d ago 👏 579 💬 10

In Predict by How to Win

MCKINSEY JUST DROPPED THEIR 2025 AI REPORT. HERE IS THE...

The top 10 takeaways of the shiny new McKinsey Report on a technology promising...

✦ Nov 21 👏 1.5K 💬 62

In Towards AI by Adham Khaled

The Day Google Made Coding Obsolete: How Gemini 3.0 and...

Don't just read another "Gemini 3 changes everything" story. See the real numbers, the...

✦ Nov 22 👏 2.1K 💬 32