AI Renaissance Weekly Newsletter

Neural Networks: Explained

AI Renaissance Issue #1

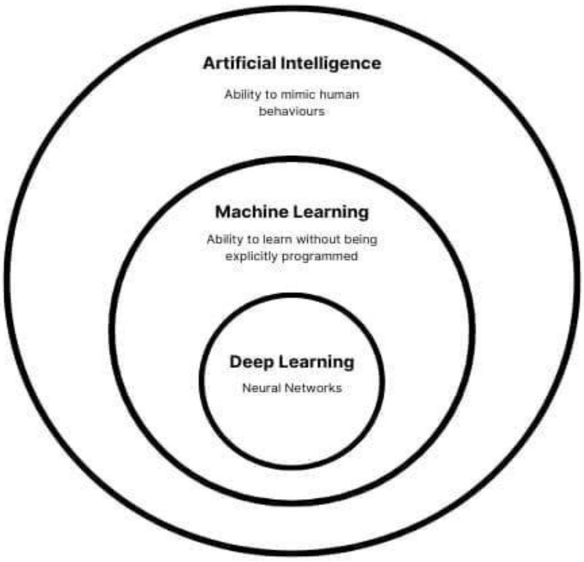
Introduction:

In this first issue of **AI Renaissance**, we dive into the fascinating world of neural networks, the backbone of modern AI advancements. From their origin in 1944 to their revolutionary applications today, neural networks have evolved through breakthroughs in deep learning. In simple terms, neural networks learn to perform tasks by analyzing examples, making them invaluable in fields like medical diagnostics and image recognition. Join us as we break down the basics of how neural networks work, including their structure, training process, and essential functions. Get ready to unlock the potential of one of AI's most versatile tools!

In the past decade, the best AI systems (such as image generators, or a machine translation service), have been using a technique called **deep learning**.

- Deep learning is an approach to AI using **neural networks**.
  So how did neural networks get so popular?

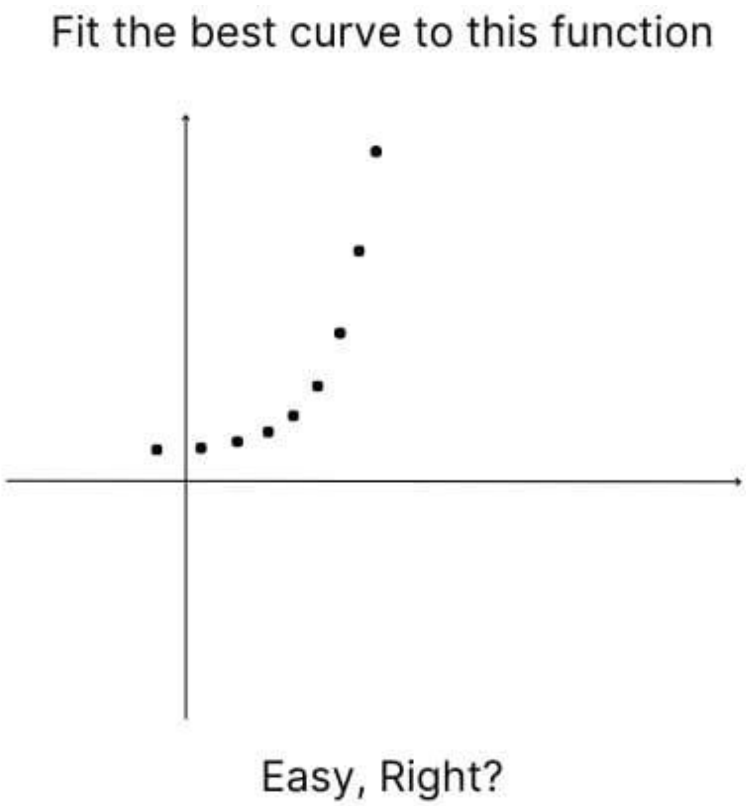
Here's a simple timeline of neural networks:



- Neural networks were first proposed in **1944** by Warren McCullough and Walter Pitts, two University of Chicago researchers.
- Neural networks were a major area of research in both **neuroscience and computer science** but have had a rough career: dying then resurging, then dying again, but finally
resurging fueled by the increased processing power of graphics cards.

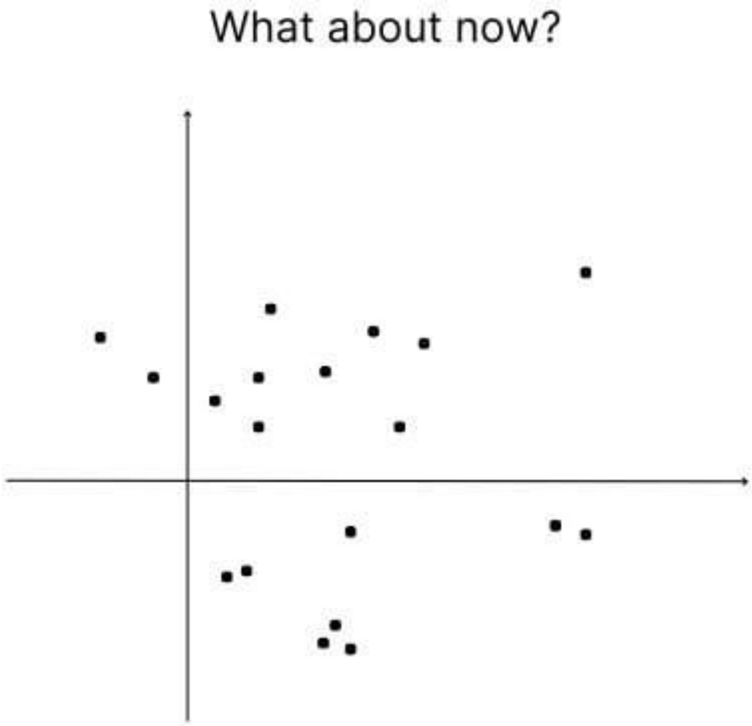
Neural networks are a very versatile tool. They are a means of **learning how to perform a task by analyzing training examples.**

Let's say you have a function. If you are given this function explicitly, then it is 100% guaranteed to replicate the correct output, given any input.



But what if you weren't given the function at all, and instead, some **data points** and wanted to guess what the function was shaped like?

For instance, there is no function that can tell you whether a tumor is malignant or benign. But using neural networks, we can **give it some data and tell it to make some sense of it**.

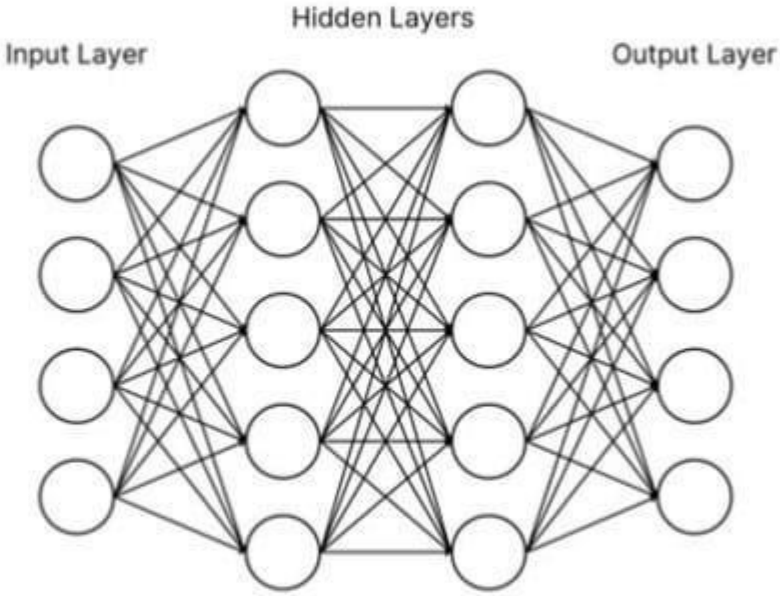


If we train it well, then it can predict whether a given tumor is malignant or benign with decent precision.

The **data** fed into a neural network will be different for different use cases. For instance, in an object recognition system, the network might be fed thousands of labeled images of cars, houses, and other objects.

Neural networks are extremely interesting, since they are modeled after a **human brain**.

It consists of many that are densely interconnected, organized in layers:



Many of today's neural networks are **feed-forward**, meaning that they pass data only forward.

To each of its incoming connections, a node will assign a number known as a "**weight**."

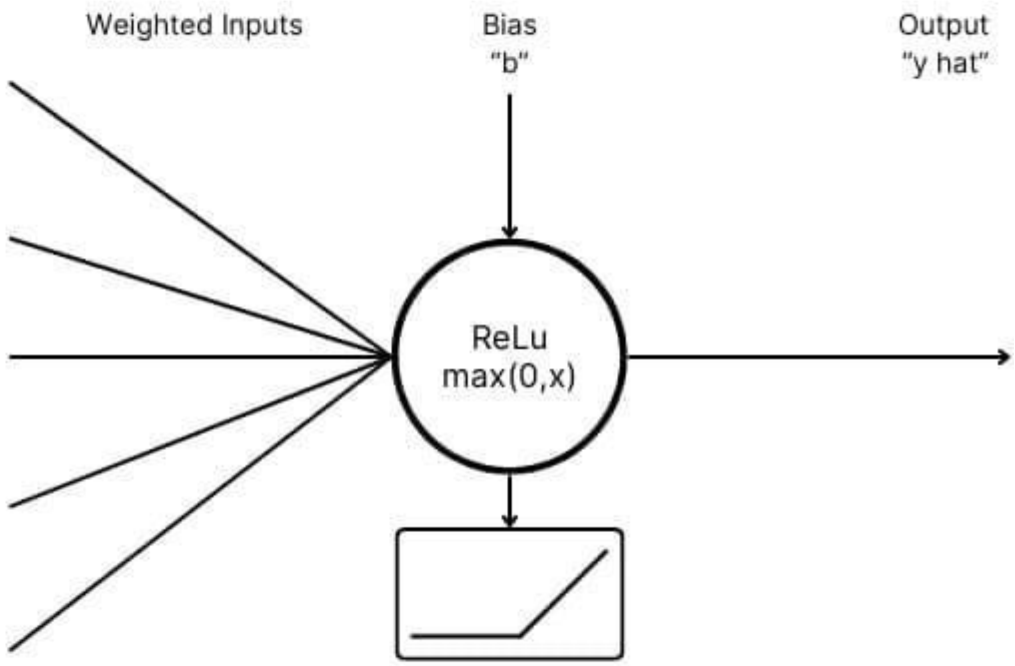
When the network is active, the node receives a different input (just a number) over each of its connections and multiplies it by the associated **weight**.

In this process, you add an additional **bias** value, that can be tweaked during training (will go into depth in next weeks' letter!).

It then adds the resulting products together, yielding a single number.

The number is put through an **activation function**, which controls the influence of that node's data on the rest of the network.

- It is used to add some **non-linearity** into the network, since always letting the data feed forward would result in a linear approximation.
- As an example, if you were to use a "**ReLU**" activation function, It would allow you to only pass data from certain neurons in a network (see example below)



The node then sends the data — the sum of the weighted inputs — along its outgoing **connections**.

When a neural net is being trained, all of its weights and thresholds are initially set to **random values**.

- Nowadays, researchers are testing certain initial values to **optimize** the learning process.

Training data is fed to the first layer, the **input layer**, and it passes through the succeeding layers, getting multiplied and added together in complex ways, until it finally arrives, at the **output layer**.

During training, the **weights** and **biases** are continually adjusted until a desired result.

- This usually means at a low enough **loss** to provide consistent results for a certain use case.
    - neural network performs
    - It's how much the network's output differs from the correct one

And that's a **simple** overlook of a neural network.

Of course, there will be more complicated concepts that can be added on, but for now, this is **generally** how a neural network works.

That's it for the first issue of AI Renaissance. See you next week!

—AI Renaissance team