

# Neural networks

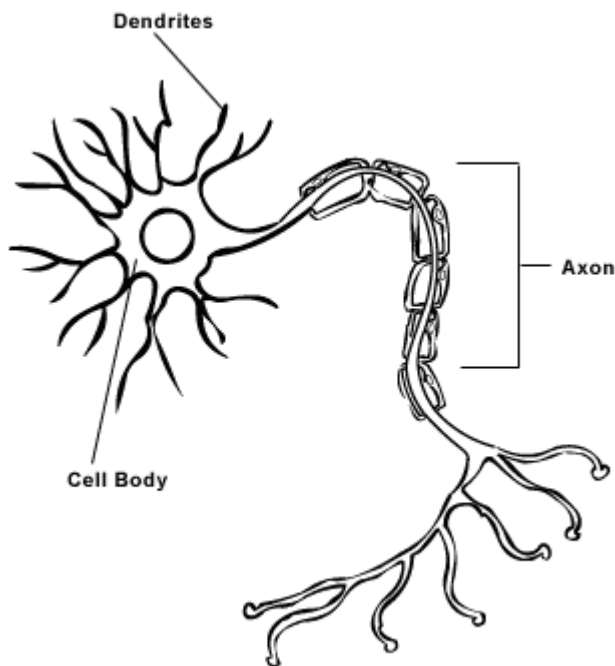
- 
- 
- 

by [Chris Woodford](#). Last updated: November 19, 2014.

Which is better—computer or brain? Ask most people if they want a brain like a [computer](#) and they'd probably jump at the chance. But look at the kind of work scientists have been doing over the last couple of decades and you'll find many of them have been trying hard to make their computers more like brains! How? With the help of neural networks—computer programs assembled from hundreds, thousands, or millions of artificial brain cells that learn and behave in a remarkably similar way to human brains. What exactly are neural networks? How do they work? Let's take a closer look!

Photo: Computers and brains have much in common, but they're essentially very different. What happens if you combine the best of both worlds—the power of a computer and the amazing flexibility of a brain? You get a superbly useful neural network. Photo of a brain scan courtesy of National Institute on Drug Abuse and [National Institutes of Health \(NIH\)](#) with overlaid neural network by explainthatstuff.com.

## How brains differ from computers

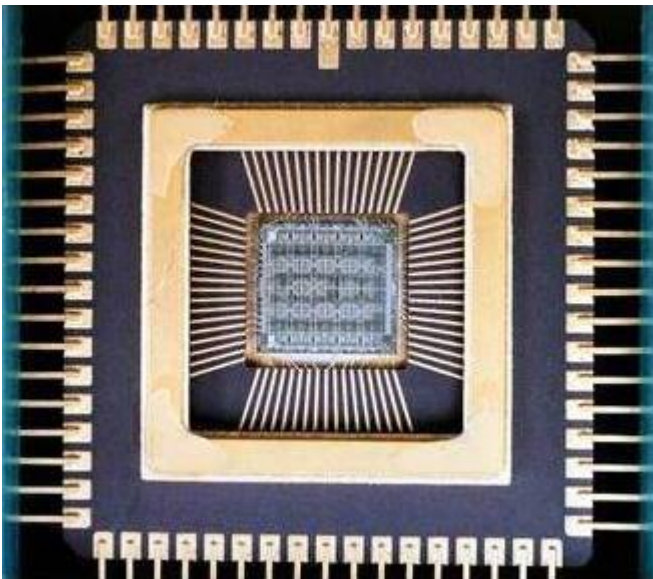


You often hear people comparing the human brain and the [electronic](#) computer and, on the face of it, they do have things in common. A typical brain contains something like 100 billion miniscule cells called neurons (no-one knows exactly how many there are and estimates go from about 50 billion to as many as 500 billion). Each neuron is made up of a cell body (the central mass of the cell) with a number of connections coming off it: numerous dendrites (the cell's inputs—carrying information toward the cell body) and a single axon (the cell's output—carrying information away). Neurons are so tiny that you could pack about 100 of their cell bodies into

a single millimeter. (It's also worth noting, briefly in passing, that neurons make up only 10 percent of all the cells in the brain; the rest are [glial cells](#), also called neuroglia, that support and protect the neurons and feed them with energy that allows them to work and grow.) Inside a [computer](#), the equivalent to a brain cell is a [nanoscopically](#) tiny switching device called a [transistor](#). The latest, cutting-edge microprocessors (single-chip computers) contain over 2 billion transistors; even a basic microprocessor has about 50 million transistors, all packed onto an [integrated circuit](#) just 25mm square (smaller than a postage stamp)!

Artwork: A neuron: the basic structure of a brain cell, showing the central cell body, the dendrites (leading into the cell body), and the axon (leading away from it). Courtesy of National Institute on Drug Abuse and [National Institutes of Health \(NIH\)](#).

That's where the comparison between computers and brains begins and ends, because the two things are completely different. It's not just that computers are cold metal boxes stuffed full of binary numbers, while brains are warm, living, things packed with thoughts, feelings, and memories. The real difference is that computers and brains "think" in completely different ways. The transistors in a computer are wired in relatively simple, serial chains (each one is connected to maybe two or three others in basic arrangements known as [logic gates](#)), whereas the neurons in a brain are densely interconnected in complex, parallel ways (each one is connected to perhaps 10,000 of its neighbors).



This essential structural difference between computers (with maybe a few hundred million transistors connected in a relatively simple way) and brains (perhaps 10–100 times more brain cells connected in richer and more complex ways) is what makes them "think" so very differently. Computers are perfectly designed for storing vast amounts of meaningless (to them) information and rearranging it in any number of ways according to precise instructions (programs) we feed into them in advance. Brains, on the other hand, learn slowly, by a more roundabout method, often taking months or years to make complete sense of something really complex. But, unlike computers, they can spontaneously put information together in astounding new ways—that's where the human creativity of a Beethoven or a Shakespeare comes from—recognizing original patterns, forging connections, and seeing the things they've learned in a completely different light.

Wouldn't it be great if computers were more like brains? That's where neural networks come in!

Photo: Electronic brain? Not quite. Inside, a typical computer chip (the central square in this artwork) is made from thousands, millions, or perhaps even a couple of billion tiny [electronic](#) switches called transistors, but

there are far fewer of them than there are cells in the human brain. Photo by courtesy of [NASA Glenn Research Center \(NASA-GRC\)](#).

## What is a neural network?

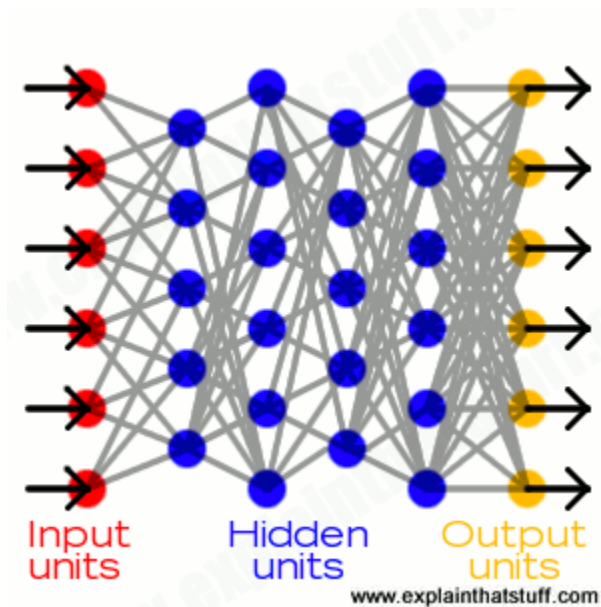
The basic idea behind a neural network is to simulate (copy in a simplified but reasonably faithful way) lots of densely interconnected brain cells inside a computer so you can get it to learn things, recognize patterns, and make decisions in a humanlike way. The amazing thing about a neural network is that you don't have to program it to learn explicitly: it learns all by itself, just like a brain!

But it isn't a brain. It's important to note that neural networks are (generally) software simulations: they're made by programming very ordinary computers, working in a very traditional fashion with their ordinary transistors and serially connected logic gates, to behave as though they're built from billions of highly interconnected brain cells working in parallel. No-one has yet attempted to build a computer by wiring up transistors in a densely parallel structure exactly like the human brain. In other words, a neural network differs from a human brain in exactly the same way that a [computer model](#) of the weather differs from real clouds, snowflakes, or sunshine. Computer simulations are just collections of algebraic variables and mathematical equations linking them together (in other words, numbers stored in boxes whose values are constantly changing). They mean nothing whatsoever to the computers they run inside—only to the people who program them.

### Real and artificial neural networks

Before we go any further, it's also worth noting some jargon. Strictly speaking, neural networks produced this way are called artificial neural networks (or ANNs) to differentiate them from the real neural networks (collections of interconnected brain cells) we find inside our brains. You might also see neural networks referred to by names like connectionist machines (the field is also called connectionism), parallel distributed processors (PDP), thinking machines, and so on—but in this article we're going to use the term "neural network" throughout and always use it to mean "artificial neural network."

## What does a neural network consist of?



A typical neural network has anything from a few dozen to hundreds, thousands, or even millions of artificial neurons called units arranged in a series of layers, each of which connects to the layers on either side. Some of them, known as input units, are designed to receive various forms of information from the outside world that the network will attempt to learn about, recognize, or otherwise process. Other units sit on the opposite side of the network and signal how it responds to the information it's learned; those are known as output units. In between the input units and output units are one or more layers of hidden units, which, together, form the majority of the artificial brain. Most neural networks are fully connected, which means each hidden unit and each output unit is connected to every unit in the layers either side. The connections between one unit and another are represented by a number called a weight, which can be either positive (if one unit excites another) or negative (if one unit suppresses or inhibits another). The higher the weight, the more influence one unit has on another. (This corresponds to the way actual brain cells trigger one another across tiny gaps called synapses.)

Photo: A fully connected neural network is made up of input units (red), hidden units (blue), and output units (yellow), with all the units connected to all the units in the layers either side. Inputs are fed in from the left, activate the hidden units in the middle, and make outputs feed out from the right. The strength (weight) of the connection between any two units is gradually adjusted as the network learns.

## How does a neural network learn things?

Information flows through a neural network in two ways. When it's learning (being trained) or operating normally (after being trained), patterns of information are fed into the network via the input units, which trigger the layers of hidden units, and these in turn arrive at the output units. This common design is called a feedforward network. Not all units "fire" all the time. Each unit receives inputs from the units to its left, and the inputs are multiplied by the weights of the connections they travel along. Every unit adds up all the inputs it receives in this way and (in the simplest type of network) if the sum is more than a certain threshold value, the unit "fires" and triggers the units it's connected to (those on its right).



For a neural network to learn, there has to be an element of feedback involved—just as children learn by being told what they're doing right or wrong. In fact, we all use feedback, all the time. Think back to when you first learned to play a game like ten-pin bowling. As you picked up the heavy ball and rolled it down the alley, your brain watched how quickly the ball moved and the line it followed, and noted how close you came to knocking down the skittles. Next time it was your turn, you remembered what you'd done wrong before, modified your movements accordingly, and hopefully threw the ball a bit better. So you used feedback to compare the outcome you wanted with what actually happened, figured out the difference between the two, and used that to change what you did next time ("I need to throw it harder," "I need to roll slightly more to the left," "I need to let go later," and so on). The bigger the difference between the intended and actual outcome, the more radically you would have altered your moves.

Photo: Bowling: You learn how to do skillful things like this with the help of the neural network inside your brain. Every time you throw the ball wrong, you learn what corrections you need to make next time. Photo by Jon Dasbach courtesy of [US Navy](#).

Neural networks learn things in exactly the same way, typically by a feedback process called backpropagation (sometimes abbreviated as "backprop"). This involves comparing the output a network produces with the output it was meant to produce, and using the difference between them to modify the weights of the connections between the units in the network, working from the output units through the hidden units to the input units—going backward, in other words. In time, backpropagation causes the network to learn, reducing the difference between actual and intended output to the point where the two exactly coincide, so the network figures things out exactly as it should.

## How does it work in practice?

Once the network has been trained with enough learning examples, it reaches a point where you can present it with an entirely new set of inputs it's never seen before and see how it responds. For example, suppose you've been teaching a network by showing it lots of pictures of chairs and tables, represented in some appropriate way it can understand, and telling it whether each one is a chair or a table. After showing it, let's say, 25 different chairs and 25 different tables, you feed it a picture of some new design it's not encountered before—let's say a chaise longue—and see what happens. Depending on how you've trained it, it'll attempt to categorize the new example as either a chair or a table, generalizing on the basis of its past experience—just like a human. Hey presto, you've taught a computer how to recognize furniture!

That doesn't mean to say a neural network can just "look" at pieces of furniture and instantly respond to them in meaningful ways; it's not behaving like a person. Consider the example we've just given: the network is not actually looking at pieces of furniture. The inputs to a network are essentially binary numbers: each input unit is either switched on or switched off. So if you had five input units, you could feed in information about five different characteristics of different chairs using binary (yes/no) answers. The questions might be 1) Does it have a back? 2) Does it have a top? 3) Does it have soft upholstery? 4) Can you sit on it comfortably for long periods of time? 5) Can you put lots of things on top of it? A typical chair would then present as Yes, No, Yes, Yes, No or 10110 in binary, while a typical table might be No, Yes, No, No, Yes or 01001. So, during the learning phase, the network is simply looking at lots of numbers like 10110 and 01001 and learning that some mean chair (which might be an output of 1) while others mean table (an output of 0).

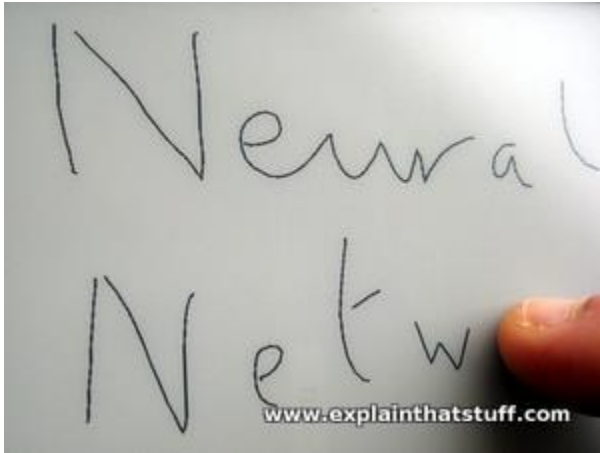
## What are neural networks used for?

On the basis of this example, you can probably see lots of different applications for neural networks that involve recognizing patterns and making simple decisions about them. In [airplanes](#), you might use a neural network as a basic autopilot, with input units reading signals from the various cockpit instruments and output units modifying the plane's controls appropriately to keep it safely on course. Inside a factory, you could use a neural network for quality control. Let's say you're producing clothes washing [detergent](#) in some giant, convoluted chemical process. You could measure the final detergent in various ways (its color, acidity, thickness, or whatever), feed those measurements into your neural network as inputs, and then have the network decide whether to accept or reject the batch.

There are lots of applications for neural networks in security, too. Suppose you're running a bank with many thousands of credit-card transactions passing through your computer system every single minute. You need a quick automated way of identifying any transactions that might be fraudulent—and that's something for which a neural network is perfectly suited. Your inputs would be things like 1) Is the cardholder actually present? 2) Has



a valid PIN number been used? 3) Have five or more transactions been presented with this card in the last 10 minutes? 4) Is the card being used in a different country from which it's registered? —and so on. With enough clues, a neural network can flag up any transactions that look suspicious, allowing a human operator to investigate them more closely. In a very similar way, a bank could use a neural network to help it decide whether to give loans to people on the basis of their past credit history, current earnings, and employment record.



Many of the things we all do everyday involve recognizing patterns and using them to make decisions, so neural networks can help us out in zillions of different ways. They can help us forecast the stockmarket or the weather, operate [radar](#) scanning systems that automatically identify enemy aircraft or ships, and even help doctors to diagnose complex diseases on the basis of their symptoms. There might be neural networks ticking away inside your computer or your [cellphone](#) right this minute. If you use cellphone apps that [recognize your handwriting](#) on a [touchscreen](#), they might be using a simple neural network to figure out which characters you're writing by looking out for distinct features in the marks you make with your fingers (and the order in which you make them). Some kinds of [voice recognition software](#) also use neural networks. And so do some of the email programs that automatically differentiate between genuine emails and [spam](#).

All in all, neural networks have made computer systems more useful by making them more human. So next time you think you might like your brain to be as reliable as a computer, think again—and be grateful you have such a superb neural network already installed in your head!

Photo: Handwriting recognition on a touchscreen, tablet computer is one of many applications perfectly suited to a neural network. Each character (letter, number, or symbol) that you write is recognized on the basis of key features it contains (vertical lines, horizontal lines, angled lines, curves, and so on) and the order in which you draw them on the screen. Neural networks get better and better at recognizing over time.

<http://www.explainthatstuff.com/introduction-to-neural-networks.html>