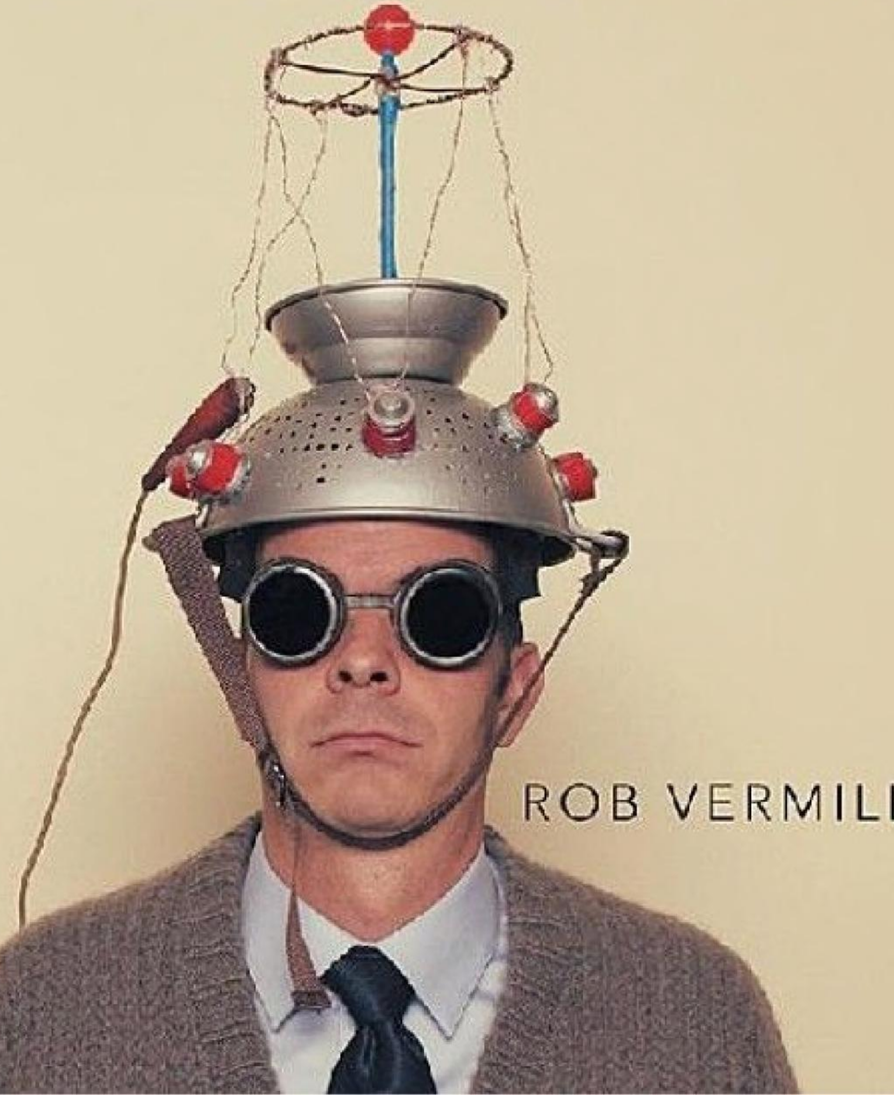
PROGRAMMER'S GUIDE TO THE BRAIN

WITH EXAMPLES IN PYTHON



ROB VERMILLER

ISBN: 978-1-4834-0001-3 (sc)
ISBN: 978-1-4834-0000-6 (e)

Library of Congress Control Number: 2019909566

Lulu Publishing Services rev. date: 08/29/2019

INTRODUCTION

Imagine we could program the brain. How cool would that be! We could lift our mood, raise our ambition, cure anxiety, bolster our self-confidence, improve our leadership skills, and alter our worst impulses. We could also create new, artificially intelligent companions and transfer our minds to powerful robot bodies.

So what's stopping us? First, we don't really know how the brain works. I believe our current understanding of brain function needs a fundamental revision. We can't program something we don't understand. Second, current theories of artificial intelligence (AI) and neural networks—based on our misunderstanding of the mind—are also cast into doubt.

I'm a computer scientist, not a neuroscientist or academic. But I know enough to say that AI—as currently conceived—is not how the brain works. So, in this book, I will intrepidly offer my own, perhaps fanciful, hopefully thought-provoking, alternative—something we can program.

So what's wrong with our current understanding of the brain?

We know the human brain contains around 100 billion neurons with many more supporting cells. Each neuron is connected to thousands of other neurons via synapses. Each neuron sends messages (electrical pulses) at a frequency up to 200 times per second to its direct neighbors. Scientists continue to devise clever lab experiments to study how human subjects think and behave. They poke and prod the brain and scan it using functional magnetic resonance imaging (fMRI) and other technologies to uncover its secrets

(Le Bihan 2014).

Yet what have we learned? Do we know how memory works? No. Do we understand how traits like ambition, shyness, fear, or risk-taking are implemented in the brain? No. Do we know why happiness “feels” happy? No. Do we know why some people are narcissists or extroverts and others are not? No. Do we know the algorithms we use to identify a potential mate? What really happens in the brain of an ambitious person? How we make plans, learn faces, take risks, and experience awe? No, no, and no.

Instead, here’s how a typical scientist describes the workings of the brain:

> The dopamine system is more or less obsessed with keeping us alive. It constantly scans the environment for new sources of food, shelter, mating opportunities, and other resources that will keep our DNA replicating … Dopamine yields not just desire but also domination. It gives us the ability to bend the environment and even other people to our will. (Lieberman and Long 2018)

I disagree. I think describing the brain in terms of molecules (e.g., neurotransmitters like dopamine) completely misses the point. A molecule is simply a bit of matter, a puff of smoke. It doesn’t know anything about food, shelter, or mating. A molecule is not an algorithm or set of instructions. If we gaze at a picture of a serotonin molecule, do we get motivated? No. Are we visually stimulated at the sight of a hormone molecule such as testosterone or estrogen? Not in the least!

You can’t program a molecule.

Molecules are simply messengers—smoke signals—helping to convey orders to an army of ready neurons in the brain. Like a general's command to charge, the signal itself conveys very little information. Much more interesting is how the receivers of the signal—the soldiers—are trained, what maneuvers they can perform, and the history and usage of their weapons. A general's order simply unleashes a complex process that's already in place. A molecule can't affect us unless our brains are prewired to be affected.

In addition to molecular explanations, neuroscientists also describe the brain in terms of its neural circuits and functional regions:

> Pride, shame, and guilt all activate similar neural circuits, including the dorsomedial prefrontal cortex, amygdala, insula, and the nucleus accumbens. Interestingly, pride is the most powerful of these emotions at triggering activity in these regions—except in the nucleus accumbens, where guilt and shame win out. This explains why it can be so appealing to heap guilt and shame on ourselves—they're activating the brain's reward center. (Korb 2015)

Again, I think this explanation misses the mark.

Yes, it's true that the brain is divided into specialized regions such as the prefrontal cortex, amygdala, and cerebellum. Planning seems to occur somewhere in the brain's frontal lobes. Emotion appears to bubble up from the amygdala, a specialized region consisting of around 12 million neurons. Short-term memory is enabled by the hippocampus. The brain stem is responsible for basic life functions and respiration. The cerebellum helps with coordinated motor control. The cortex—or gray matter—comprises around 20 percent of the brain's neurons and appears responsible for language, vision, and other higher-order capabilities.

But knowing which brain regions are more active when we engage in a specific activity doesn't help us understand how the mind is implemented. Describing the brain in terms of its gross anatomy and specialized regions is no more helpful than explaining it in terms of neurotransmitters and hormones.

You can't program a brain region.

Yet scientists continue to plunge ahead with their current approaches. The latest $100 million scientific research project called MICrONS (Cepelewicz 2016) endeavors to understand the brain by studying a cubic millimeter of a rat's brain tissue—containing 100,000 neurons and one billion synapses—in the visual cortex, the part of the brain involved in sight. Best of luck to them, but I'm not holding my breath. A similar $1.3 billion Human Brain Project, launched by the European Union in 2013, collapsed after only two years (Theil 2015).

To truly understand how the brain and AI work, I propose that we focus on the activity of individual neurons. They don't even have to be human neurons. The Open Worm project (openworm.org) studies a small nematode worm having only 300 neurons. Other scientists conduct research on large sea slugs—*Aplysia californica*—that have 20,000 central neurons in their nervous system, still a manageable number. Focusing on small worms and sea slugs is much more practical and has a much better chance of success in allowing us to understand how the brain works in general. Why? Because in small worms and sea slugs, many or all of the identified neurons have a unique function and carry out the same task across the species (Hoyle and Wiersma 1977).

Like sea slugs and nematodes, I speculate that human brains also contain identified neurons, each having a specific task to perform, although this is not scientifically proven. More radically, I think each neuron is responsible for the same function across the human species.

Think about it for a moment. If we all have the same set of identified neurons, it would explain a lot about how the brain works. Each neuron is accountable for performing a single, specific mental trait or task. Each neuron essentially becomes a computer on a network, able to run algorithms and programs, store information, and send messages to other neurons on the network. Neurons bootstrap with programs and algorithms written in our ancestral DNA. This is very similar to how the internet works today. Why not human brains?

Identified neurons give us something we can program!

When describing this theory of identified neurons, I'll deliberately ignore the influence of other, commonly ascribed causes for human traits such as genetics, epigenetics, neurotransmitters, and hormones. I believe a neuron-centric view can replace them all.

Any complete theory of the mind should also address where "feeling" and consciousness come from. I propose that the activity of individual neurons and their algorithms, in the context of powerful emergent effects, is the only explanation needed. The brain's simulation of reality, through the action of billions of individual neurons, begins to resonate in harmony, and "feeling" and consciousness are the result.

Many of these hypotheses may seem a little far-fetched, and in truth they are highly speculative and unproven. According to a critic:

> Here is what we are *not* born with: information,

> data, rules, software, knowledge, lexicons, representations, algorithms, programs, models, memories, images, processors, subroutines, encoders, decoders, symbols, or buffers—design elements that allow digital computers to behave somewhat intelligently. Not only are we not born with such things, we also don't develop them—ever. (R. Epstein 2016)

Clearly, I disagree. The ideas in this book are falsifiable, and that's how science advances (Popper 1959). Science is a way of thinking, not a source of absolute truth. Will these ideas ultimately prove to be correct? Perhaps yes, perhaps no. But I think anything is better than the current quagmire of our understanding.

Finally, I'll discuss the implications of this theory and how algorithms, trait diversity, and luck can answer persistent questions about free will, personal responsibility, and fairness. I'll address the social issues arising from AI and how we can best prepare society for the changes to come.

It's a lot to cover, so let's get started!

PART 1—IDENTIFIED NEURONS

AI Is Not How the Brain Works

Artificial intelligence (AI) is a hot topic these days, offering the promise of self-driving cars and trucks, automated factories, and world-class chess play. Yet the enthusiasm is also laced with foreboding: *Will a robot take my job?*

Current AI techniques are known by many names, including “neural networks” and “deep learning,” a subset of machine learning (LeCun, Bengio and Hinton 2015). AI is used to mine big data sets to look for patterns, predict weather trends, diagnose complex diseases, monitor banking transactions for fraud, identify military targets, forecast stock market trends, convert speech to text, translate languages, enable factory quality control, understand handwriting, and—perhaps ominously—parse social media for consumer and political preferences.

AI can also be trained to recognize faces in photos and videos as we often see on TV police dramas. When a criminal culprit’s face is caught on camera, it doesn’t take long for law enforcement to identify him or her using AI facial recognition tools. AI is very good at finding patterns in big data, especially when millions or even billions of exemplars are available to train the neural network (Loy 2019).

However, as promising as AI might appear for these specific applications, dire warnings of an “AI winter” are mounting, as current approaches to AI don’t explain how the brain actually works.

> To get a deep-learning system to recognize a hot dog, we might have to feed it 40 million pictures of hot dogs. To get [a girl] to recognize a hot dog, we show her a [single] hot dog …
>
> A computer that sees a picture of … doughnuts piled up on a table and captions it, automatically, as “a pile of doughnuts piled on a table” seems to understand the world; but when that same program sees a picture of a girl brushing her teeth and says “The boy is holding a baseball bat” we realize how thin that understanding really is …
>
> Self-driving cars can fail to navigate conditions they’ve never seen before. Machines have trouble parsing sentences that demand common-sense understanding of how the world works. (Somers 2017)

We’re often willing to suspend our disbelief and accept that true AI and intelligent robots are right around the corner. We see them in the movies all the time, and they seem credible and real. But in reality, they are still a ways off.

NYU psychology and neuroscience professor Gary Marcus thinks AI—as currently conceived—is fragile, inflexible, and unable to generalize or think outside the box. AI can’t think abstractly, make plans, or comprehend complex ideas (Marcus 2018a). Today, AI can merely recognize patterns. It can’t currently do what humans achieve effortlessly. It lacks common sense. Neural networks are missing the innate prewiring required to learn.

One interviewer described Marcus’s thinking:

> Say you wanted a machine to teach itself to recognize daisies … At first, the neural network is just guessing blindly; it starts life a blank slate, more or less … Given enough time and enough daisies, the neural net gets more accurate …
>
> But Marcus was never convinced. For him, the problem is the blank slate: It assumes that humans build their intelligence purely by observing the world around them, and that machines can too. But Marcus doesn't think that's how humans work. He walks the intellectual path laid down by Noam Chomsky, who argued that humans are born wired to learn, programmed to master language and interpret the physical world. (Thompson 2018)

We humans easily learn new concepts after a single example or exposure. We can generalize from a few examples. We effortlessly infer cause and effect. We pick up language quickly.

So how is the brain prewired to learn this way?

Scientists have long believed that human memories are stored in the connections, or synapses, between neurons in the brain. In 1949, Donald Hebb proposed that the connection weights between neurons are adjusted as we learn skills and remember new things. This process, known as "backpropagation," is "a procedure for rejiggering the strength of every connection in the network so as to fix the error for a given training example" (Somers 2017). It's the basis of today's AI and deep-learning artificial neural networks.

But recent research suggests other mechanisms at work. Our memories are not always stored in the connections between neurons but, instead, are stored within individual neurons themselves (Bédécarrats, et al. 2018). When scientists disrupted neural synapses in snail brains in a way that should have removed their memory of previously administered electric shocks, some memory remained.

In another experiment, when certain RNA molecules (similar to DNA) were transferred from the brain cells of trained snails to those of untrained snails, some of the trained snails' training (memory) was transferred too. This finding—that memories may be stored in RNA and are transferable—flies in the face of established theories that memories are stored via changes in connection strengths between neurons.

Taken together, these experiments don't offer a grand refutation of current brain theory and AI, but they do show chinks in the armor of the current paradigm. Although AI and neural networks offer powerful pattern recognition capabilities, they don't reflect how the brain actually works.

Since 2012, when it was used to win the prestigious ImageNet competition (Krizhevsky, Sutskever and Hinton 2017), the neural network approach to AI has become all the rage. But prior to that, AI researchers combined multiple techniques, including symbolic rule-engines, autonomous agents, simulation, statistics, and even Kalman filters (Russell and Norvig 2009).

Today, we're at an inflection point again, as researchers realize that neural networks are fragile, don't explain their decisions well, lack common sense, and require an overwhelming amount of labelled data to train them (Marcus 2018b). When the current paradigm begins to crumble, we need a new theory to replace it.

The brain is a complex adaptive system, with properties like unpredictability, non-periodic behavior, feedback loops, spontaneous order, adaptation and emergence (Holland

2006). To study how the brain actually works, I propose we return to earlier AI techniques, especially agent-based simulation (Abar, et al. 2017), whereby neurons are autonomous agents, and the mind emerges from their collective activity in a grand simulation of reality.

That's what I intend to demonstrate. Fancifully, of course!

Neurons as Little Computers

I propose that individual identified neurons are the primary actors (agents) in the brain. They each have their own unique agenda, and they zealously and relentlessly strive to achieve their goals and objectives. In the process, they cooperate and compete with other neurons in the brain for access to scarce bodily resources and attention.

Why do I propose this? For one thing, other complex networks—such as the public internet—take this distributed approach. Human society operates this way as well. The collective activity of each individual's self-interested behavior has unintended benefits—the "invisible hand" (Smith 1759)—that central planning could never equal.

In the brain, each of the 100 billion neurons is a specialized agent, an autonomous computer executing a unique program to carry out its assigned tasks. The most basic computer is called a Turing machine, described by Alan Turing in 1936. In its simplest form, all a computer needs to operate is a long tape (onto which it can write characters) and an interpreter which can process the current symbol (or instruction) on the tape. I speculate that each neuron is just such a computer. It's not a mainstream hypothesis, to be sure, but a few intrepid scientists have proposed that microtubules inside neurons can indeed support quantum computing (Hameroff and Penrose 2013).

A computer program processes inputs and delivers outputs. It repeatedly executes a set of instructions (loop), makes conditional assessments (if–then–else), and stores data (memory). That's what brain activity is mostly about, I propose.

A computer program is simply an algorithm:

> For many people, the word "algorithm" evokes the arcane and inscrutable machinations of big data … hardly a source of practical wisdom or guidance for human affairs. But an algorithm is just a finite sequence of steps used to solve a problem … Long before algorithms were ever used by machines, they were used by people …
>
> When we think about computers, we think about coldly mechanical, deterministic systems; machines applying rigid deductive logic … [However] tackling real world tasks requires being comfortable with chance, trading off time with accuracy, and using approximations. As computers become better tuned to real-world problems, they provide not only algorithms that people can borrow for their own lives, but a better standard against which to compare human cognition itself. (Christian and Griffiths 2016)

In this book, I will de-emphasize artificial neural networks in favor of a more algorithmic approach, whereby neurons act as independent agents, running local programs in a massively distributed, resonating mind simulation. Although neural networks can also be considered algorithms, in some respects, I believe their black-box decision-making process remains too inscrutable for evolved traits. To the degree we humans employ machine learning and neural networks in the brain, I propose they execute within individual neurons themselves as helper functions.

So let's have a look at what the neural programming might look like.

Introduction to Neural Programming

The idea of a computer program running inside each of the 100 billion neurons in the brain might sound far-fetched. So take a deep breath, suspend your disbelief, and assume for a moment it's true. What are the implications?

Every program requires a programming language, a language of thought if you will. In neurons, the language must be ancient, written hundreds of thousands, or even millions, of years ago. Since we don't know what a neuron's programming language actually looks like, I'll use a modified version of the modern Python language in this book to illustrate the examples. (Python aficionados may protest some liberties I've taken with the language. I apologize in advance.)

Those with a computer programming background will know that most programming languages are essentially the same. They have subroutines or methods, variable assignments (A = 3), conditionals (if–then–else), arrays and lists (1, 2, 3, 4, 5), mathematical functions (A + B), and loops.

By my theory, each human trait is carried out by a single identified neuron in the brain. So let's assume there's a `SeekShelterNeuron` in the brain that's responsible for our behavior to find shelter when we're exposed to the elements like rain or snow. It seems like a reasonable hardwired behavior as we humans—and monkeys before us—have sought shelter throughout our history.

The neuron’s programming—using a modern agent-based (object-oriented) style that encapsulates code into classes—might look something like this:

```
class SeekShelterNeuron:
  def _init_(self):
    # Initialize the neuron once,
    # when it’s first generated in the brain.
    self.exposed = False

  # Then automatically run this main routine at
  startup
  def run(self):
    # Repeat (or loop) the following code forever,
    # from birth to death of the neuron
    loop forever:
      # send a message to another neuron
      # to delegate a subtask or make a request
      # e.g., are we currently exposed to the
      elements?
      e = sendrequest(“ExposedNeuron”, “getstatus”)
      # assign the resulting status to a variable
      self.exposed = e
      # then check whether the result is True or
      False
      if e == True:
        # The conditional statement (IF) is True, so
        # do something, e.g., move our legs and arms
        … code not shown
      else:
        # otherwise do something else
        … code not shown
      wait(clocktick) # wait a moment, rinse and repeat
    # Repeat the loop
```

The `SeekShelterNeuron` executes a unique program, which

I’ve encapsulated into a class of the same name. (I’ll assume there’s only a single instance neuron per class; that’s why I will treat classes and instances interchangeably for now.)

The class contains two methods or subroutines: `_init_` and `run`. The `_init_` method is executed only once, when the neuron is first generated in the brain, at or before birth. The `run` method then executes automatically after that. It usually contains a loop that repeatedly executes the code that follows it from the birth to the death of the neuron. In other words, every neuron in the brain is actively running a program all the time—none of this “we only use 10 percent of our brain” stuff.

A neuron may delegate subtasks to other neurons. In the previous example, a request or message is sent to the `ExposedNeuron` to determine our current state (e.g., whether we’re currently exposed to the elements and needing shelter). [Narrowly-defined delegated subtasks like this may indeed employ local pattern matching and machine learning techniques.] A conditional (if–then–else) statement handles the two possibilities—True or False—likely by delegating additional subtasks to other neurons. The loop then waits briefly and restarts from the beginning.

To recap, the neural programming language supports the following operations:

- **Assignment of local variables** (using the [=] operator). In the example, the variable *e* is assigned a value: True or False.
- **Test for equality** (using the [==] operator). In the example, *e* is tested to see whether it equals True or False. “Not equal to” is denoted by the [!=] operator.
- **Conditional statements** (if–then–else). These allow the neuron to exhibit different behaviors depending on

external or internal inputs or context.

- **Loops.** The “loop forever” statement endlessly repeats the execution of a block of code that follows it for the life of the neuron. In the example, a request is sent to the `ExposedNeuron` every clock tick, waiting for a `True` reply. Most of the time, the neuron sits idly because we’re not in an exposed situation (e.g., sitting in rain or cold). The neuron just keeps waiting—pinging—until events transpire.
- **Sending messages or requests.** A neuron can send a request to any other neuron in the brain using the `sendrequest` method. The destination or address of the message—that is, the neuron’s name—is in quotation marks because the referenced neuron may be far away from the sender and may not even currently exist in the brain. Requests (and replies) take time to process and are routed from one neuron to the next until they reach their destination via other neurons that are already busy with their own tasks. Until the message is received, the reply is assumed to be `False`. Once received, it is locally cached.
- **Comments in the code.** These are preceded by a hash mark (#), and the program ignores them. Like a hastily written computer program, our DNA code is probably not commented very well for easy deciphering, but we can always hope!

I propose that this theory of identified neurons running code written (by evolution) in an agent-based (object-oriented) computer language, with integrated message passing between neurons, provides all we need to explain how the brain works. When billions of identified neurons simultaneously seek to carry out their own unique agendas, in cooperation and competition with other neurons, this massively parallel simulation begins to resonate with reality and takes on a life of its own.

The simulation aspect comes from how the neuron is programmed. After executing its short programming script, the neuron waits a clock tick, observes how the world has changed (via messages from other neurons), adjusts its internal state, then repeats the same script, over and over again, like a chess player repeatedly reacting to his opponent's moves, or an economist engaged in game theory. The mind emerges from this rich simulation of reality.

Loading the Program (from DNA)

For a neuron to act as a computer, it must first load its unique program. So where's the code? Why not in our shared human DNA library?

First a word about DNA, or "deoxyribonucleic acid." Each cell in the human body, including every neuron in the brain, contains a complete copy of the DNA library. The DNA library resembles a scroll—two meters long—containing 3 billion nucleic acids or "bases." [How a two-meter-long DNA molecule can be stuffed into each cell in the body is mind-boggling! But anyway, every neuron in the brain has access to the complete DNA library.]

There are four letters in the DNA alphabet—A, T, G, and C—represented by the four types of bases, adenine, thymine, guanine, and cytosine. This alphabet is comparable to the 26-letter English alphabet or the two-letter binary alphabet of 1's and 0's used in computers. It can represent concepts, words, and descriptions. And computer programs.

The DNA library contains 23 books or volumes (called chromosomes), onto which are written a total of 3 billion letters or characters, one letter per base. (Technically, we receive one set of chromosomes from our mother and another set from our father, so we have 46 chromosomes.)

Now a word about genes. All humans have the same 20,000 genes. Genes are books or chapters of the DNA library used to describe and express proteins and enzymes, which construct the body and control metabolism. (Technically, genes act as templates for constructing proteins. DNA bases are grouped into triples, or codons—ATT, GAC, and so on—which represent amino acids. ATG stands for methionine, TGG stands for tryptophan, and so on. Genes are thus translated into amino acid sequences, a.k.a. "proteins.")

Now the complication. Only 9 percent of our DNA is dedicated to genes (of which 2 percent acts as protein templates and another 7 percent is involved in gene regulation.) In other words, genes comprise only 9 percent of our DNA. So what does the other 91 percent of our non-protein-coding, non-gene DNA do?

Some scientists call it "junk DNA." I speculate that this junk DNA is actually a storage device for programs executed by neurons in the brain. Each neuron is assigned a unique section of junk DNA from which it loads and executes its program. In the rest of this book, I will ignore genes and genetics and will focus exclusively on the other 91 percent of junk DNA.

It's just a wild theory, but it does make sense. If neurons run ancient computer programs, then the programs have to be located somewhere. Why not in our collective human DNA? (To prove such a theory, perhaps a computer scientist can run a decompiler on our junk DNA to look for code fragments.)

Obviously, the neural code isn't written in English. It would be compiled into four-letter DNA-ese, similar to how modern computer code is compiled into a binary (base-2) alphabet.

The programming code for each identified neuron is stored as a sequence of letters in one of the library books of our collective human DNA:

`ATTGATCGGCAATGACTTAAGGGCACCGAT … and so on`

We humans are all 99.9 percent alike, DNA-speaking. We're so similar because at one point in our evolutionary history, as we left Africa, there were perhaps as few as 5,000 humans left on earth. Our DNA similarity has a stunning implication. If neurons load their programs from our collective human DNA library, it follows that we must all share the same neural programs! All humans have the same set of identified neurons running the same algorithms! It must be so because the 0.1 percent DNA difference among us is not enough to swap in and out different sections of DNA. We all share the same DNA.

DNA is a wormhole to the time of our ancestors. Millions of years of their experience is manifest in DNA, which transports that history to the present. DNA is time travel (Buonomano 2017). DNA is merely matter, but it's a different sort of matter than, say, a rock. A rock lives in the eternal present—a fleeting series of nows. It doesn't retain any memory of the past—lights, sounds, tastes, smells, touches, causes and effects. Over millions of years, we evolved from rocks, and water and gas, and life began to retain knowledge from the past. First, matter took on the shape of catalysts, which sped the rate of chemical reactions. Then, matter plus catalysts constructed more and more complex structures with feedback loops. Catalysts created more catalysts, along with more sophisticated structures like DNA, which began to replicate. More and more knowledge was manifest in the form of matter, and experience was retained and transported from the past to the present. Our neural programming—in our DNA—allows ancestral memories to resonate with present experience.

Not everyone agrees, however, that evolution can devise such elegant solutions as identified neurons and neural programs. Consider the following, for example:

> How genetics and development actually work, it's a mess. It consists entirely of hacks and patches all the way down. It's not modular. It's not agile. It's not anything that an engineer would recognize; it's just crap that runs. So when you go to try to reverse-engineer it, you can't. It's no good, because it was never engineered in the first place. So how do you devolve what has been evolved? That's like trying to unstir the coffee. (Veve 2018)

I disagree. Obviously humans can design things, and humans are merely a product of evolution. It follows that evolution can also design things. Indeed, intelligent design—and elegant solutions—may be an inherent function of evolution itself, or of any complex adaptive system.

Let me repeat that. Anything we humans can do—from setting up small Skunk Works labs (local innovation), to venture capital investment (scale-up), to radical redesign of current approaches (deployment)—can also be done by evolution because we can do it and we're simply a product of evolution.

Neurons Send Each Other Messages

A neuron doesn't have to do everything itself. It can delegate subtasks and solicit help from thousands of other neurons by sending them request messages. Those neurons can then delegate their sub-subtasks to other neurons, and so on. Pretty soon, the whole brain is lit up in a cascade of activity initiated by a single neuron. It's no wonder people think the brain operates holistically. The mind is the product of billions of individual neurons—self-interested and self-directed agents—each cooperating and competing for resources.

In the brain, each neuron is physically connected to “only” a few thousand other neurons. (The axon of one neuron directly connects to the dendrites of thousands of other neurons.) In order to send a request or message to one of its immediate neighbors, a neuron simply converts its request into a series of electrical pulses—like Morse code—and transmits the pulses via its axon wire to their dendrites.

The first time a `sendrequest` message is sent from one neuron to another, there will likely be no immediate response. The request will simply “time out,” and the reply will be assumed to be `False`. However, after some time has elapsed, the asynchronous reply will eventually come back. Once it’s received, it will be locally cached in the requesting neuron for better performance the next time it’s needed.

Caching of results—like squirrels storing nuts for the winter—is critical to the operation of a massively parallel architecture like the brain. Even if the memory storage capability of the brain were infinite, we are limited by the time it takes to search for things and send messages to other neurons because of constraints on network bandwidth in the brain. It makes sense that neurons cache the results of messages into local working memory, inside the neuron itself, although that cache size is limited. Computer scientists have long debated the best strategy for cache management, including first in, first out (FIFO) and least recently used (LRU) (Christian and Griffiths 2016). It turns out that LRU is the most efficient way to manage the cache: message results that haven’t been used in a while—like the stalest nuts in a squirrel’s cache—are flushed from the local cache.

How can a neuron send a message to a distant neuron in the brain if the two are not physically connected? In computer science, there’s a communication protocol called a “message passing interface” (MPI) whereby messages are sent to remote computers. Operating on a similar concept, neural messages could be routed from one neuron to the next until they reach their destination, either by point-to-point communication or broadcast messaging to many neurons simultaneously.

For example, on the public internet, packets of data are routed from one server to the next until they reach their destination. Likewise, each busy neuron in the brain—by this highly speculative theory—has a second job: to forward messages. To accomplish this complex task, each neuron maintains a “routing table” containing instructions on how to best route messages to remote nodes. (Routing tables can be built dynamically through trial and error by observing network traffic, or with the assistance of “routing protocols.”)

```
class SomeNeuron:
  def _init_(self):
    # initialize the message routing information
    self.routing_table = []
    self.immediate_neighbors = []

  def routerequest(self, requestor, reqtype, dest,
  params):
    # find the best routing for the message
    if dest == self:
      # process the request locally
      sendrequest(self, requestor, reqtype, params)
    else:
      rt = self.routing_table
      neighbors = self.immediate_neighbors
      via = determine_best_routing(dest, neighbors, rt)
```

```
        sendrequest(destination, reqtype, params,
        next=via)
```

The `self.routing_table` and `self.immediate_neighbors` are arrays of values stored in the neuron as local variables (i.e., local memory).

With routing, neurons can't expect to get an immediate reply to their `sendrequest` messages from other helper neurons and delegates. Network latency may be an issue, and responses may lag, as anyone who plays games or streams content on the internet can tell you.

In this book, I'll assume that replies from other neurons arrive nearly instantaneously, but I realize this is an unrealistic expectation.

Neurons Have Unique Names

Aplysia californica are large sea slugs that graze underwater in tidal zones. When threatened, they release ink into the water to confuse their predators. Each slug has a tongue on its underside controlled by two neurons. When the slug is provoked, its siphon and gill can be quickly retracted (Moroz 2011).

Every aplysia has 20,000 central neurons in its nervous system. Many of the neurons are unique and carry out the same specialized function across the species.

Are we humans like sea slugs? Do we have unique identified neurons in our brain, each with its own task and identity? Do all humans have the same identified neurons?

Yes, that's my proposal, although it's a radical and as-yet-unproven theory. There's a single neuron in the human brain for, say, hunger, a single neuron that implements greed or ambition. Every neuron in the human brain has a specific purpose, a specific set of tasks—goals and objectives—to perform. The human mind emerges from the collective activity of individual, empowered neurons.

Why do I propose this? There are three primary reasons:

1. Evolution is lazy and conservative. If something is good enough for sea slugs, it's good enough for us humans. Evolution wouldn't reinvent the way the brain functions from earlier species unless there was a very good reason for doing so, and I don't think there is one.
2. Humans are defined by their evolved traits, fears, passions, interests, motivations, and drives. Does the brain implement each of these traits in a different way? Not likely. Evolution likely found a common approach for all of them. Identified neurons offer just such a powerful explanatory framework.
3. Neurons can't send messages to other neurons—for example, to delegate tasks—unless they know their name (or address) in advance, *a priori*.

This last point is crucial. Consider, again, the public internet. Each computer on the internet has a unique IP address that allows any other computer in the world to send it messages. Without a unique IP address, modern technologies like instant messaging, chat, and streaming video wouldn't be possible.

It's the same with neural communications. Each neuron must know, in its ancient programming, the name/address of every other neuron in the brain it wishes to correspond with. Without knowing, *a priori*, the names of all other neurons in the brain, there would be no way for one neuron to send a message to another neuron. (I'll discuss an exception to this rule in a moment.)

Using the proposed neural programming language described earlier, here's how a neuron might send a message to another neuron:

```
hunger_status = sendrequest("HungerNeuron", "detect")
```

In this example, the name of the `HungerNeuron` is known in advance by the ancient evolutionary code that references it. The task to determine (detect) whether we're hungry is delegated to this neuron by sending it a request. When the reply is received, it's stored in a local variable called `hunger_status`.

Because we humans all share 99.9 percent identical DNA, we must all share the same identified neurons. As new neurons are generated in the brain, they are assigned unique names from the DNA name directory. Once a neuron has a unique name, it bootstraps by loading and running its unique program, also located in the DNA library. That code then executes the neuron's agenda over and over, for a lifetime.

You may have spotted a flaw with the idea of uniquely naming each neuron. Cumulatively, our DNA is only 3 billion letters long. That's not enough DNA to store billions of unique names for each of the 100 billion neurons in the brain!

So let's modify the theory. Many neurons in the brain may be closely related clones. For example, the retina at the back of the eye contains millions of nearly identical neurons, each of them running the same program. Likewise, the visual cortex in the back of the brain, where visual processing takes place, also contains a large array of identical neurons. It's probable that each of these identical neurons is assigned a unique name algorithmically—for example, `RetinaNeuron-1234`—by adding a random number to the end of the neuron's class name when the neuron is generated. It can still be sent messages to its unique name/address, but as its name is not known until runtime (i.e., after birth), it must be registered upon generation by the `GenesisNeuron` if others are to be able to look it up.

Neurogenesis

After conception, the brain develops rapidly in the womb. Billions of new neurons are generated every day. Some scientist believe that process stops at birth, and we're born with all the neurons we'll ever receive (Sorrells, et al. 2018). But others have disputed this, contending that we continue to generate new neurons into old age, especially in the region of the brain known as the hippocampus (Boldrini 2018). I'll assume that either new neurons are generated into adulthood, or they are pre-generated (pre-allocated) earlier for later use.

New neurons are generated through a process called neurogenesis. Neurons connect physically—via dendrites and axons—to a few thousand of their nearest neighbors. That process establishes a basic physical network connectivity of the brain. Before neurons receive a unique name, they can only send broadcast messages over the physical network. Once they receive a unique name (address), they can send narrowcast messages to each other.

If each mental trait is carried out by a single identified neuron, what happens when the neuron dies? Does it lose all its data and memories? Perhaps, as with modern data centers, each neuron periodically archives its memories to a remote storage device, likely a remote DNA backup tape. Then, when the neuron is regenerated (respawned), its memories can be restored by following a disaster recovery protocol.

Let’s assume the existence of a single `GenesisNeuron` in the brain that acts as a factory to generate new neurons:

```
class GenesisNeuron:
  def _init_(self):
    # start life with an empty list of new neurons
    self.neuron_list = []

  def sendrequest(self, requestor, reqtype, parameters):
    # this method responds to requests from other neurons
    # to generate a new neuron
    if reqtype == “generate”:
      # generate a new neuron
      if parameters != False:
        neuron_class = parameters
      else:
        neuron_class = “GenericNeuron”
      new_neuron = New(neuron_class)
      self.neuron_list.append(new_neuron)
      return new_neuron
    elsif reqtype == “inventory”:
      return self.neuron_list

  def run(self):
    # ensure all neurons listed in the DNA directory
    # exist, i.e., are generated or re-generated
    loop forever:
      directory = read_neuron_directory_from_DNA()
      loop for n in directory:
        e1 = sendrequest(n, “ping”)
        wait(clocktick) # wait a moment
        e2 = sendrequest(n, “ping”)
        if e1 == False and e2 == False:
          # The neuron didn’t respond, so either
```

```
                # it doesn’t exist yet, or it died.
                # [Disaster recovery protocol is not shown]
                # In either case, generate a new neuron
                sendrequest(self, “generate”, n)
        wait(clocktick)
    # repeat loop
```

Neurons are generated either at the request of other neurons (perhaps to represent new concepts, as we’ll see later) or automatically from the predefined DNA directory of neural names. Did I mention this is all highly speculative?

The `sendrequest` method of the `GenesisNeuron` processes requests to generate a new neuron. The new neuron is given a name prefix as an input parameter from the requester, or simply `GenericNeuron`. The newly created neuron is then added to the `self.neuron_list`, which can obviously get quite long.

Neurons listed in the innate DNA directory are automatically generated or regenerated by the `run` method. It continually loops through the predefined neural names in the directory and pings them to see if they exist. If not, it creates them.

(Python programmers might complain that there’s no `New` function in the language to create an instance of an arbitrary class. But that’s not what’s actually happening here. The `New` function spins up a new computer—that is, each neuron gets its own virtual machine—gives it a unique name [e.g., `GenericNeuron-1234`], adds it to the network, loads its program from our shared DNA library, and starts the program running for a lifetime.)

There’s One Neuron per Trait

Each human trait—desire, motivation, want, need, impulse, interest, or passion—is implemented by a single identified messages with other neurons to carry out its tasks.

So, what became of hormones, neurotransmitters, and neuropeptides? They're still there, of course, in the brain. But I believe that such molecules have been relegated to a relatively minor role. Now they simply serve to make the transmission of neural messages more efficient. Instead of having a neuron send individual messages to thousands of other neurons, broadcast messages can be sent via hormones and neurotransmitters. Like the general who shouts "charge!" to initiate a battle, it's more effective to send up a single smoke signal for distant troops to see and comply with, rather than communicate with each of them individually using messages.

In other words, molecules like serotonin, dopamine, testosterone, and estrogen are simply used for network optimization. Hormones and neurotransmitters don't cause happiness or depression or risk-taking or trust behaviors. They simply optimize the network of messages sent between neurons. Individual neurons have evolved to become the primary implementers of our traits and behaviors, and molecules now play a secondary and supporting role.

PART 2—MEMORY

The “iPad Neuron”

Some scientists believe a new neuron is generated in the brain every time we learn a new concept, even suggesting that a “grandmother neuron” exists in the brain (Bowers 2009). Since grandmothers have existed for millions of years of evolutionary history, it’s certainly possible that the “grandmother” concept is itself innate.

But what about modern objects that didn’t exist in the state of nature? How do they get into our brains? Take iPads for example. They weren’t invented until recently, so there’s little chance evolution has had time to incorporate that memory engram into our collective human DNA library. Is there an “iPad neuron” in the brain, and if so, how did it get there?

Let’s take a step back. Why bother to remember an iPad at all? Because it interests us and motivates us, we crave it and covet it, and we envy our friends who own one. Without a strong emotion associated with an object, we simply can’t remember it. We wouldn’t want to remember it.

When we see another person gazing or staring intently at an iPad and then smiling, it triggers a feeling of envy in us toward the object. Bingo! Now it becomes meaningful to us. If we can associate an evolved emotion—for example, envy—to an object, then it’s worth remembering:

```
class EnvyNeuron:
  def __init__(self):
     # begin life with no objects of envy
```

https://www.nytimes.com/2019/02/07/opinion/trump-socialism-state-of-the-union.html.

Le Bihan, D. 2014. *Looking Inside the Brain: The Power of Neuroimaging.* Princeton University Press.

LeCun, Y, Y Bengio, and G Hinton. 2015. “Deep learning.” *Nature* 521 (7553), 436.

Libet, B, C Gleason, E Wright, and D Pearl. 1983. “Time of Conscious Intention to Act in Relation to Onset of Cerebral Activity (Readiness-Potential).” *Brain* 106 (3): 623–42. doi:10.1093/brain/106.3.623. PMID 6640273.

Lieberman, D, and M Long. 2018. *The Molecule of More: How a Single Chemical in Your Brain Drives Love, Sex, and Creativity—and Will Determine the Fate of the Human Race.* BenBella Books.

Loy, J. 2019. *Neural Network Projects with Python: The ultimate guide to using Python to explore the true power of neural networks through six projects.* Packt Publishing. https://towardsdatascience.com/how-to-build-your-own-neural-network-from-scratch-in-python-68998a08e4f6.

Mahler, J. 2015. “The White and Gold (No, Blue and Black!) Dress That Melted the Internet.” *New York Times.* https://www.nytimes.com/2015/02/28/business/a-simple-question-about-a-dress-and-the-world-weighs-in.html.

Marcus, G. 2018a. *Deep Learning: A Critical Appraisal.* Jan 2. https://arxiv.org/abs/1801.00631.

—. 2018b. *Innateness, AlphaZero, and Artificial Intelligence.* Jan 17. https://arxiv.org/abs/1801.05667.

Mayford, M, S Siegelbaum, and E Kandel. 2012. “Synapses and Memory Storage.” *Cold Spring Harb Perspect Biol.* doi:10.1101/cshperspect.a005751.

McAuliffe, K. 2019. “Liberals and Conservatives React in Wildly Different Ways to Repulsive Pictures.” *Atlantic*, Mar. https://www.theatlantic.com/magazine/archive/2019/03/the-yuck-factor/580465/.

Michotte, A. 1946/1963. *The Perception of Causality.* NY: Basic Books (translation).

Mitchell, K. 2018. *Innate: how the wiring of our brains shapes who we are.* Princeton Univ Press.

Moroz, L. 2011. “Aplysia.” *Current Biology* 21(2): R60–R61. doi:10.1016/j.cub.2010.11.028.

NIH. 2014. “Hearing Different Frequencies.” *NIH.* Jun 2. https://www.nih.gov/news-events/nih-research-matters/hearing-different-frequencies.

Pinedo, M. 2016. *Scheduling: Theory, Algorithms, and Systems.* Springer.

Pinker, Steven. 1994. *The Language Instinct.* Penguin.

Plutchik, R. 2003. *Emotions and life: Perspectives from psychology, biology, and evolution.* Washington, DC: American Psychological Association.

Popper, Karl. 1959. *The Logic of Scientific Discovery.* Routledge.

Queenan, B. 2017. “On the research of time past: the hunt for the substrate of memory.” *Annals of the New York Academy of Sciences* 1396: 108-125. doi:10.1111/nyas.13348.

Quiroga, R Q, L Reddy, G Kreiman, C Koch, and I Fried. 2005. “Invariant visual representation by single neurons in the human brain.” *Nature*, Jun: 1102-1107.

Rawls, J. 1971. *A Theory of Justice.* Harvard University Press.

Russell, S, and P Norvig. 2009. *Artificial Intelligence: A Modern Approach (3rd Edition).* Pearson.

Salam, M, and D Victor. 2018. “Yanny or Laurel? How a Sound Clip Divided America.” *New York Times.* https://www.nytimes.com/2018/05/15/science/yanny-laurel.html.

Schenk, D. 2011. *The Genius in All of Us: New Insights into Genetics, Talent, and IQ.* Anchor.

Schopenhauer, A. 1839. *On The Freedom Of The Will.* as translated in The Philosophy of American History : The Historical Field Theory (1945) by Morris Zucker, p. 531.

Shomrat, T, and M Levin. 2013. “An automated training paradigm reveals long-term memory in planarians and its persistence through head regeneration.” *The Journal of Experimental Biology* 216: 3799-3810. doi:10.1242/jeb.087809.

Smith, A. 1759. *The theory of moral sentiments.* London: Printed for A. Millar, and A. Kincaid and J. Bell.

Snyder, B. 2005. *Save The Cat! The Last Book on Screenwriting You'll Ever Need.* Michael Wiese Productions.

Solem, J E. 2012. *Programming Computer Vision with Python: Tools and algorithms for analyzing images.* O'Reilly Media.

Somers, J. 2017. "Is AI Riding a One-Trick Pony?" *MIT Technology Review*, Nov/Dec.

Sorrells, S, M Paredes, A Cebrian-Silla, and et al. 2018. "Human hippocampal neurogenesis drops sharply in children to undetectable levels in adults." *Nature* 555, pages 377–381. doi:10.1038/nature25975.

Theil, S. 2015. "Why the Human Brain Project Went Wrong—and How to Fix It." *Scientific American.* https://www.scientificamerican.com/article/why-the-human-brain-project-went-wrong-and-how-to-fix-it/.

Thompson, C. 2018. "The Miseducation of Artificial Intelligence." *WIRED*, Dec.

Veve, A. 2018. "Farsighted: Stewart Brand, the man who was wired before WIRED, on the tools he believes will make the whole world better." *WIRED*, Oct.

Wade, N. 2007. *Before the Dawn: Recovering the Lost History of Our Ancestors.* Penguin.

Welch, J, and S Welch. 2006. "It's Not about Empty Suits." *Business Week*, Oct 16.

—. 2008. "Release Your Inner Extrovert." *Business Week*, Nov 26.

Wikipedia. n.d. *The Apprentice.* Mark Barnett. https://en.wikipedia.org/wiki/The_Apprentice_(American_TV_series).

—. n.d. *Wason Selection Task.* https://en.wikipedia.org/wiki/Wason_selection_task.

Yang, A. 2018. *The War on Normal People: the truth about America's disappearing jobs and why universal basic income is our future.* Hachette Books.