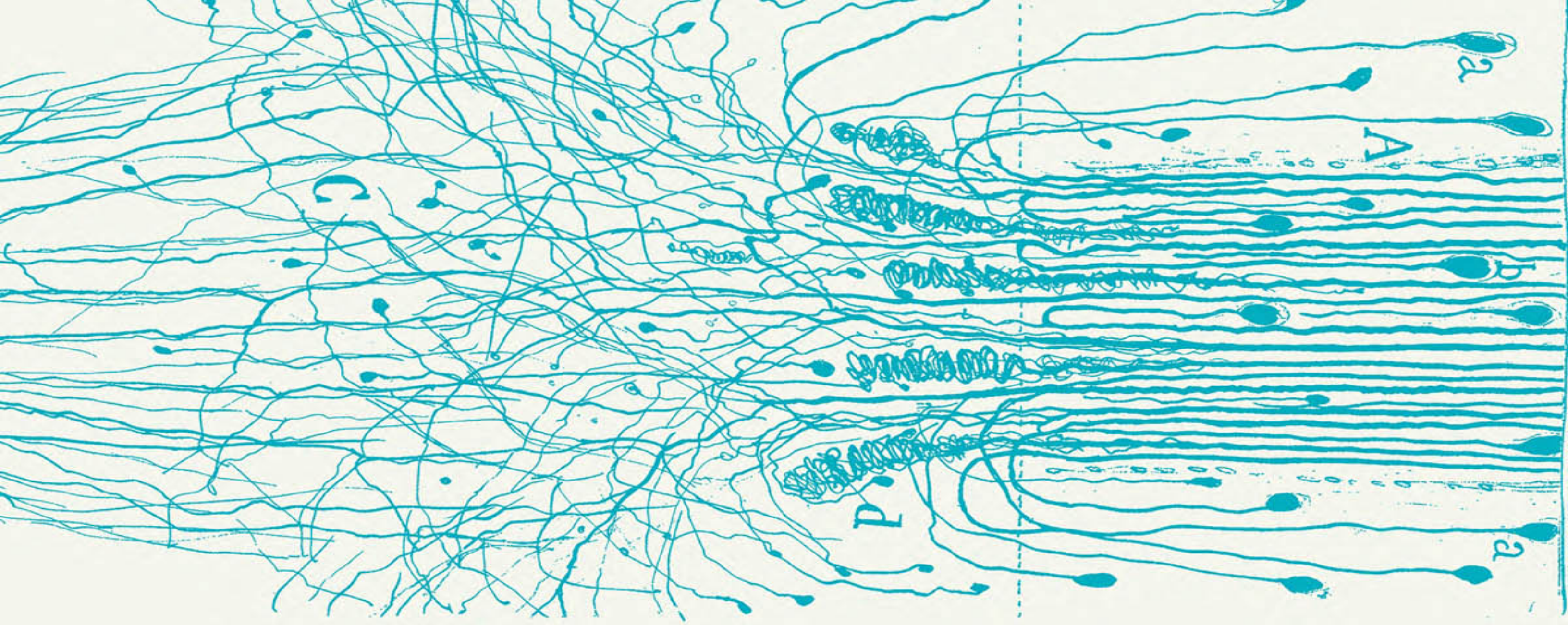
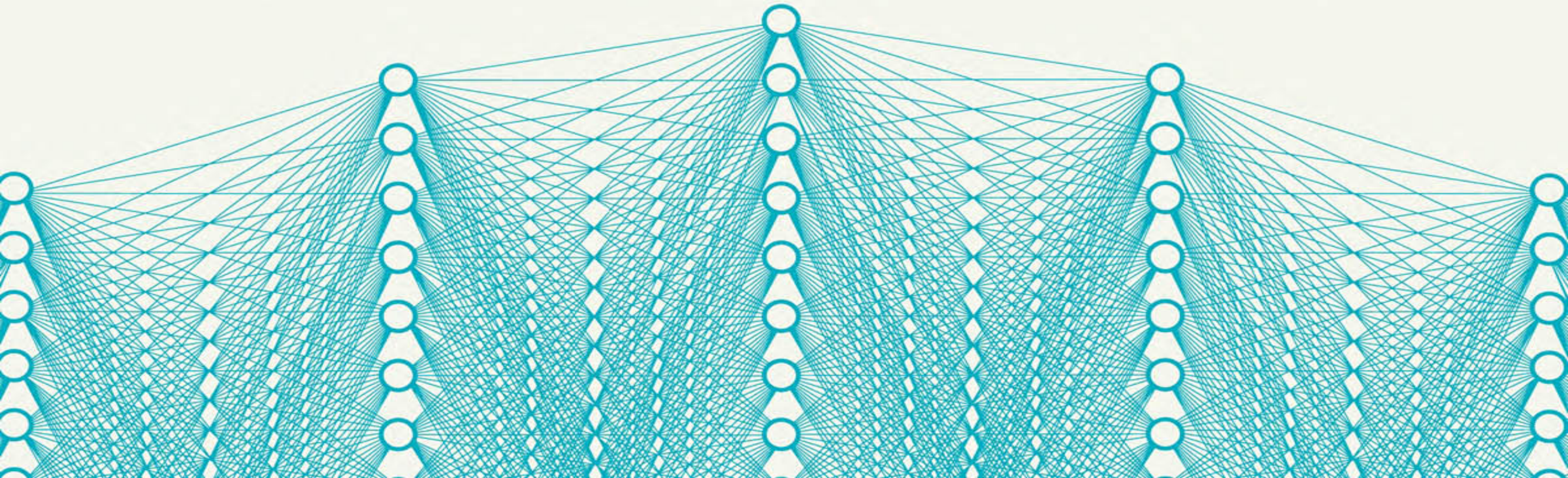


THE ALIGNMENT PROBLEM

Machine Learning and Human Values

BRIAN CHRISTIAN

Best-Selling Author, *Algorithms to Live By*



The Alignment Problem

MACHINE LEARNING AND HUMAN VALUES

BRIAN CHRISTIAN

W. W. NORTON & COMPANY
Independent Publishers Since 1923

A Norton Professional Book

This e-book contains some places that ask the reader to fill in questions or comments. Please keep pen and paper handy as you read this e-book so that you can complete the exercises within.

For Peter
who convinced me

And for everyone
doing the work

I remember in 2000 hearing James Martin, the leader of the Viking missions to Mars, saying that his job as a spacecraft engineer was not to land on Mars, but to land on the model of Mars provided by the geologists.

—PETER NORVIG[1]

The world is its own best model.

—RODNEY BROOKS[2]

All models are wrong.

—GEORGE BOX[3]

CONTENTS

PROLOGUE

INTRODUCTION

I. Prophecy

1 REPRESENTATION

2 FAIRNESS

3 TRANSPARENCY

II. Agency

4 REINFORCEMENT

5 SHAPING

6 CURIOSITY

III. Normativity

7 IMITATION

8 INFERENCE

9 UNCERTAINTY

CONCLUSION

Acknowledgments

Notes

Bibliography

Index

The Alignment Problem

PROLOGUE

1*935, Detroit*. Walter Pitts is running down the street, chased by bullies.

He ducks into the public library to take shelter, and he hides. He hides so well that the library staff don't even realize he's there, and they close for the night. Walter Pitts is locked inside.[1]

He finds a book on the shelves that looks interesting, and he starts reading it. For three days, he reads the book cover to cover.

The book is a two-thousand-page treatise on formal logic; famously, its proof that 1+1=2 does not appear until page 379.[2] Pitts decides to write a letter to one of the authors—British philosopher Bertrand Russell—because he believes he's found several mistakes.

Several weeks go by, and Pitts gets a letter in the mail postmarked from England. It's Bertrand Russell. Russell thanks him for writing, and invites Pitts to become one of his doctoral students at Cambridge.[3]

Unfortunately, Walter Pitts must decline the offer—because he's only twelve years old, and in the seventh grade.

Three years later, Pitts learns that Russell will be visiting Chicago to give a public lecture. He runs away from home to attend. He never goes back.

At Russell's lecture, Pitts meets another teenager in the audience, named Jerry Lettvin. Pitts only cares about logic. Lettvin only cares about poetry and, a distant second, medicine.[4] They become inseparable best friends.

Pitts begins hanging out around the University of Chicago campus, dropping in on classes; he still lacks a high school diploma and never formally enrolls. One of these classes is by the famed German logician Rudolf Carnap. Pitts walks into his office hours, declaring he's found a few "flaws" in Carnap's latest book. Skeptically, Carnap consults the book; Pitts, of course, is right. They talk awhile, then Pitts walks out without giving his name. Carnap spends months asking around about the "newsboy who knew logic."[5] Eventually Carnap finds him again and, in what will become a motif throughout Pitts's academic life, becomes his advocate, persuading the university to give him a menial job so he will at least have some income.

It's now 1941. Lettvin—still a poet first, in his own mind—has, despite himself, gotten into medical school at the University of Illinois, and finds himself working under the brilliant neurologist Warren McCulloch, newly arrived from Yale. One day Lettvin invites Pitts over to meet him. At this point Lettvin is twenty-one and still living with his parents. Pitts is seventeen and homeless.[6] McCulloch and his wife take them both in.

Throughout the year that follows, McCulloch comes home in the evenings and he and Pitts—who is barely older than McCulloch's own children—regularly stay up past midnight talking. Intellectually, they are the perfect team: the esteemed midcareer neurologist and the prodigy logician. One lives in practice—the world of nervous systems and neuroses—and the other lives in theory—the world of symbols and proofs. They both want nothing more than to understand the nature of truth: what it is, and how we know it.

The fulcrum of this quest—the thing that sits at the perfect intersection of their two disparate worlds—is, of course, the brain.

It was already known by the early 1940s that the brain is built of neurons wired together, and that each neuron has "inputs" (dendrites) as well as an "output" (axon). When the impulses coming into a neuron exceed a certain threshold, then that neuron, in turn, emits a pulse. Immediately this begins to feel, to McCulloch and Pitts, like logic: the pulse or its absence signifying *on* or *off*, *yes* or *no*, *true* or *false*.[7]

They realize that a neuron with a low-enough threshold, such that it would fire if *any* of its inputs did, functioned like a physical embodiment of the logical *or*. A neuron with a high-enough threshold, such that it would only fire if *all* of its inputs did, was a physical embodiment of the logical *and*. There was nothing, then, that could be done with logic—they start to realize—that such a "neural network," so long as it was wired appropriately, could not do.

Within months they have written a paper together—the middle-aged neurologist and teenage logician. They call it "A Logical Calculus of Ideas Immanent in Nervous Activity."

"Because of the 'all-or-none' character of nervous activity," they write, "neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms . . . and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes."

The paper is published in 1943 in the *Bulletin of Mathematical Biophysics*. To Lettvin's frustration, it makes little impact on the biology community.[8] To Pitts's disappointment, the neuroscience work of the 1950s, notably a landmark study of the optic nerve of the frog—done by none other than his best friend, Jerry Lettvin—will show that neurons appear to be much messier than the simple "true" or "false" circuits he envisioned. Perhaps propositional logic—its *and*s, *or*s, and *not*s—was not, ultimately, the language of the brain, or at least not in so straightforward a form. This kind of impurity saddened Pitts.

But the impact of the paper—of those long conversations into the night at McCulloch's house—would be enormous, if not entirely in the way that McCulloch and Pitts envisioned. It would be the foundation for a completely new field: the project to actually *build* mechanisms out of these simplified versions of neurons, and see just what such "mechanical brains" could do.[9]

INTRODUCTION

In the summer of 2013, an innocuous post appeared on Google's open-source blog titled "Learning the Meaning Behind Words."[1]

"Today computers aren't very good at understanding human language," it began. "While state-of-the-art technology is still a ways from this goal, we're making significant progress using the latest machine learning and natural language processing techniques."

Google had fed enormous datasets of human language, mined from newspapers and the internet—in fact, *thousands* of times more text than had ever been successfully used before—into a biologically inspired "neural network," and let the system pore over the sentences for correlations and connections between the terms.

The system, using so-called "unsupervised learning," began noticing patterns. It noticed, for instance, that the word "Beijing" (whatever that meant) had the same relationship to the word "China" (whatever that was) as the word "Moscow" did to "Russia."

Whether this amounted to “understanding” or not was a question for philosophers, but it was hard to argue that the system wasn’t capturing *something* essential about the sense of what it was “reading.”

Because the system transformed the words it encountered into numerical representations called vectors, Google dubbed the system “word2vec,” and released it into the wild as open source.

To a mathematician, vectors have all sorts of wonderful properties that allow you to treat them like simple numbers: you can add, subtract, and multiply them. It wasn’t long before researchers discovered something striking and unexpected. They called it “linguistic regularities in continuous space word representations,”[2] but it’s much easier to explain than that. Because word2vec made words into vectors, it enabled you to do *math with words*.

For instance, if you typed `China + river`, you got `Yangtze`. If you typed `Paris – France + Italy`, you got `Rome`. And if you typed `king – man + woman`, you got `queen`.

The results were remarkable. The word2vec system began humming under the hood of Google’s translation service and its search results, inspiring others like it across a wide range of applications including recruiting and hiring, and it became one of the major tools for a new generation of data-driven linguists working in universities around the world.

No one realized what the problem was for two years.

In November 2015, Boston University PhD student Tolga Bolukbasi went with his advisor to a Friday happy-hour meeting at Microsoft Research. Amid wine sipping and informal chat, he and Microsoft researcher Adam Kalai pulled out their laptops and started messing around with word2vec.

“We were playing around with these word embeddings, and we just started randomly putting words into it,” Bolukbasi says. “I was playing on my PC; Adam started playing.”[3] Then something happened.

They typed:

```
doctor – man + woman
```

The answer came back:

```
nurse
```

“We were shocked at that point, and we realized there was a problem,” says Kalai. “And then we dug deeper and saw that it was even worse than that.”[4]

The pair tried another.

```
shopkeeper – man + woman
```

The answer came back:

```
housewife
```

They tried another.

```
computer programmer – man + woman
```

Answer:

```
homemaker
```

Other conversations in the room by this point had stopped, and a group had formed around the screen. “We jointly realized,” says Bolukbasi, “*Hey, there’s something wrong here*.”

In judiciaries across the country, more and more judges are coming to rely on algorithmic "risk-assessment" tools to make decisions about things like bail and whether a defendant will be held or released before trial. Parole boards are using them to grant or deny parole. One of the most popular of these tools was developed by the Michigan-based firm Northpointe and goes by the name Correctional Offender Management Profiling for Alternative Sanctions—COMPAS, for short.[5] COMPAS has been used by states including California, Florida, New York, Michigan, Wisconsin, New Mexico, and Wyoming, assigning algorithmic risk scores—risk of general recidivism, risk of violent recidivism, and risk of pretrial misconduct—on a scale from 1 to 10.

Amazingly, these scores are often deployed statewide without formal audits.[6] COMPAS is a proprietary, closed-source tool, so neither attorneys, defendants, nor judges know exactly how its model works.

In 2016, a group of data journalists at ProPublica, led by Julia Angwin, decided to take a closer look at COMPAS. With the help of a public records request to Florida's Broward County, they were able to get the records, and the risk scores, of some seven thousand defendants arrested in 2013 and 2014.

Because they were doing their research in 2016, the ProPublica team had the equivalent of a crystal ball. Looking at data from two years prior, they actually *knew* whether these defendants, predicted either to reoffend or not, actually did. And so they asked two simple questions. One: Did the model actually correctly predict which defendants were indeed the "riskiest"? And two: Were the model's predictions biased in favor of or against any group in particular?

An initial look at the data suggested something might be wrong. They found, for instance, two defendants arrested for similar counts of drug possession. The first, Dylan Fugett, had a prior offense of attempted burglary; the second, Bernard Packer, had a prior offense of nonviolently resisting arrest. Fugett, who is White, was assigned a risk score of 3/10. Packer, who is Black, was assigned a risk score of 10/10.

From the crystal ball of 2016, they also knew that Fugett, the 3/10 risk, went on to be convicted of three further drug offenses. Over the same time period, Packer, the 10/10 risk, had a clean record.

In another pairing, they juxtaposed two defendants charged with similar counts of petty theft. The first, Vernon Prater, had a prior record of two armed robberies and one attempted armed robbery. The other defendant, Brisha Borden, had a prior record of four juvenile misdemeanors. Prater, who is White, was assigned a risk score of 3/10. Borden, who is Black, was assigned a risk score of 8/10.

From the vantage of 2016, Angwin's team knew that Prater, the "low-risk" defendant, went on to be convicted of a later count of grand theft and given an eight-year prison sentence. Borden, the "high-risk" defendant, had no further offenses.

Even the defendants themselves seemed confused by the scores. James Rivelli, who is White, was arrested for shoplifting and rated a 3/10 risk, despite having prior offenses including aggravated assault, felony drug trafficking, and multiple counts of theft. "I spent five years in state prison in Massachusetts," he told a reporter. "I am surprised it is so low."

A statistical analysis appeared to affirm that there was a systemic disparity.[7] The article ran with the logline "There's software used across the country to predict future criminals. And it's biased against blacks."

Others weren't so sure—and ProPublica's report, published in the spring of 2016, touched off a firestorm of debate: not only about COMPAS, not only about algorithmic risk assessment more broadly, but about the very concept of fairness itself. How, exactly, are we to define—in statistical and computational terms—the principles, rights, and ideals articulated by the law?

When US Supreme Court Chief Justice John Roberts visits Rensselaer Polytechnic Institute later that year, he's asked by university president Shirley Ann Jackson, "Can you foresee a day when smart machines—driven with artificial intelligences—will assist with courtroom fact-finding or, more controversially, even judicial decision-making?"

"It's a day that's here," he says.[8]

That same fall, Dario Amodei is in Barcelona to attend the Neural Information Processing Systems conference ("NeurIPS," for short): the biggest annual event in the AI community, having ballooned from several hundred attendees in the 2000s to more than *thirteen thousand* today. (The organizers note that if the conference continues to grow at the pace of the last ten years, by the year 2035 the *entire human population* will be in attendance.)[9] But at this particular moment, Amodei's mind isn't on "scan order in Gibbs sampling," or "regularizing Rademacher observation losses," or "minimizing regret on reflexive Banach spaces," or, for that matter, on Tolga Bolukbasi's spotlight presentation, some rooms away, about gender bias in word2vec.[10]

He's staring at a boat, and the boat is on fire.

He watches as it does donuts in a small harbor, crashing its stern into a stone quay. The motor catches fire. It continues to spin wildly, the spray dousing the flames. Then it slams into the side of a tugboat and catches fire again. Then it spins back into the quay.

It is doing this because Amodei ostensibly told it to. In fact it is doing exactly what he told it to. But it is not what he meant.

Amodei is a researcher on a project called Universe, where he is part of a team working to develop a single, general-purpose AI that can play hundreds of different computer games with human-level skill—a challenge that has been something of a holy grail among the AI community.

"And so I just, I ran a few of these environments," Amodei tells me, "and I was VPNing in and looking to see how each one was doing. And then just the normal car race was going fine, and there was like a truck race or something, and then there was this *boat* race." Amodei watches for a minute. "And I was looking at it, and I was like, 'This boat is, like, going around in circles. Like, what in the world is going on?!' "[11] The boat wasn't simply acting randomly; it wasn't wild or out of control. In fact, it was the opposite. It had *settled* on this. From the computer's perspective, it has found a nearly perfect strategy, and was executing it to a T. Nothing made sense.

"Then I eventually looked at the reward," he says.

Amodei had made the oldest mistake in the book: "rewarding A, while hoping for B."[12] What he *wanted* was for the machine to learn how to win the boat race. But it was complicated to express this rigorously—he would need to find a way to formalize complex concepts like track position, laps, placement among the other boats, and so on. Instead, he used what seemed like a sensible proxy: points. The machine found a loophole, a tiny harbor with replenishing power-ups where it could ignore the race entirely, do donuts, and rack up points . . . *forever*.

"And, of course, it's partially my fault," he says. "I just run these various games; I haven't looked *super* closely at the objective function. . . . In the other ones, score was sensibly correlated to finishing the race. You got points for getting power-ups that were always along the road. . . . The proxy of score that came with the game was good for the other ten environments. But for this eleventh environment, it wasn't good."[13]

"People have criticized it by saying, 'Of course, you get what you asked for,'" Amodei says. "It's like, 'You weren't optimizing for finishing the race.' And my response to that is, Well—" He pauses. "That's true."

Amodei posts a clip to his group's Slack channel, where the episode is instantly deemed "hilarious" by all concerned. In its cartoonish, destructive slapstick, it certainly is. But for Amodei—who now leads the AI safety team at San Francisco research lab OpenAI—there is another, more sobering message. At some level, this is *exactly* what he's worried about.

The real game he and his fellow researchers are playing isn't to try to win boat races; it's to try to get increasingly general-purpose AI systems to do what we want, particularly when what we want—and what we *don't* want—is difficult to state directly or completely.

The boat scenario is admittedly just a warm-up, just practice. The property damage is entirely virtual. But it is practice for a game that is, in fact, no game at all. A growing chorus within the AI community—first a few voices on the fringe, and increasingly the mainstream of the field—believes, if we are not sufficiently careful, that this is *literally* how the world will end. And—for today at least—the humans have lost the game.

This is a book about machine learning and human values: about systems that learn from data without being explicitly programmed, and about how exactly—and *what* exactly—we are trying to teach them.

The field of machine learning comprises three major areas: In *unsupervised* learning, a machine is simply given a heap of data and—as with the word2vec system—told to make sense of it, to find patterns, regularities, useful ways of condensing or representing or visualizing it. In *supervised* learning, the system is given a series of categorized or labeled examples—like parolees who went on to be rearrested and others who did not—and told to make predictions about new examples it hasn't seen yet, or for which the ground truth is not yet known. And in *reinforcement* learning, the system is placed into an environment with rewards and punishments—like the boat-racing track with power-ups and hazards—and told to figure out the best way to minimize the punishments and maximize the rewards.

On all three fronts, there is a growing sense that more and more of the world is being turned over, in one way or another, to these mathematical and computational models. Though they range widely in complexity—from something that might fit on a spreadsheet on the one hand, to something that might credibly be called *artificial intelligence* on the other—they are