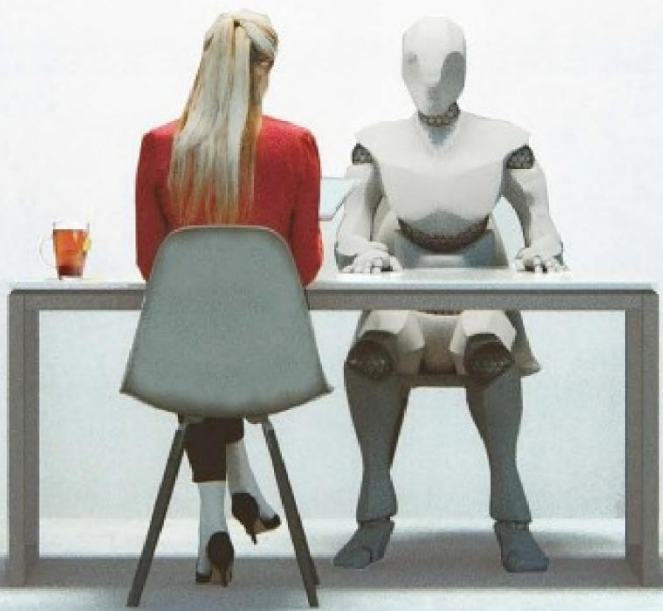
'Vital reading. This is the book on artificial intelligence that we need right now.' **Mike Krieger, co-founder of Instagram**

THE ALIGNMENT PROBLEM

How Can Artificial Intelligence Learn Human Values?



BRIAN CHRISTIAN

First published in the United States in 2020 by W. W. Norton & Company, Inc., New York.

First published in hardback in Great Britain in 2021 by Atlantic Books, an imprint of Atlantic Books Ltd.

10 9 8 7 6 5 4 3 2 1

A CIP catalogue record for this book is available from the British Library.

Hardback ISBN: 978 1 78649 430 6
Trade paperback ISBN: 978 1 78649 431 3
E-book ISBN: 978 1 78649 432 0

Printed in Great Britain

Atlantic Books
An imprint of Atlantic Books Ltd
Ormond House
26–27 Boswell Street
London
WC1N 3JZ

www.atlantic-books.co.uk

CONTENTS

THE ALIGNMENT PROBLEM

PROLOGUE

1935, 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 opensource 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 factfinding 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 steadily replacing both human judgment *and* explicitly programmed software of the more traditional variety.

This is happening not only in technology, not only in commerce, but in areas with ethical and moral weight. State and federal law increasingly mandates the use of "risk-assessment" software to determine bail and parole. The cars and trucks on our freeways and neighborhood streets are increasingly driving themselves. We no longer assume that our mortgage application, our résumé, or our medical tests will be seen by human eyes before a verdict is rendered. It is as if the better part of humanity were, in the early twenty-first century, consumed by the task of gradually putting the world—figuratively and literally—on autopilot.

In recent years, alarm bells have gone off in two distinct communities. The first are those focused on the present-day ethical risks of technology. If a facial-recognition system is wildly inaccurate for people of one race or gender but not another, or if someone is denied bail because of a statistical model that has never been audited and that no one in the courtroom—including the judge, attorneys, and defendant—understands, this is a problem. Issues like these cannot be addressed within traditional disciplinary camps, but rather only through dialogue: between computer scientists, social scientists, lawyers, policy experts, ethicists. That dialogue has begun in a hurry.

The second are those worried about the future dangers that await as our systems grow increasingly capable of flexible, real-time decisionmaking, both online and in the physical world. The past decade has seen what is inarguably the most exhilarating, abrupt, and worrying progress in the history of machine learning—and, indeed, in the history of artificial intelligence. There is a consensus that a kind of

taboo has been broken: it is no longer forbidden for AI researchers to discuss concerns of safety. In fact, such concerns have over the past five years moved from the fringes to become one of the central problems of the field.

Though there is a rivalry of sorts over whether the immediate or the longer-term issues should take priority, these two communities are united in their larger aims.

As machine-learning systems grow not just increasingly pervasive but increasingly powerful, we will find ourselves more and more often in the position of the "sorcerer's apprentice": we conjure a force, autonomous but totally compliant, give it a set of instructions, then scramble like mad to stop it once we realize our instructions are imprecise or incomplete—lest we get, in some clever, horrible way, precisely what we asked for.

How to prevent such a catastrophic divergence—how to ensure that these models capture our norms and values, understand what we mean or intend, and, above all, do what we want—has emerged as one of the most central and most urgent scientific questions in the field of computer science. It has a name: *the alignment problem*.

In reaction to this alarm—both that the bleeding edge of research is getting ever closer to developing so-called "general" intelligence and that real-world machine-learning systems are touching more and more ethically fraught parts of personal and civic life—has been a sudden, energetic response. A diverse group is mustering across traditional disciplinary lines. Nonprofits, think tanks, and institutes are taking root. Leaders within both industry and academia are speaking up, some of them for the first time, to sound notes of caution —and redirecting their research funding accordingly. The first generation of graduate students is matriculating who are focused explicitly on the ethics and safety of machine learning. The alignment problem's first responders have arrived at the scene.

This book is the product of nearly a hundred formal interviews and many hundreds of informal conversations, over the course of four years and many tens of thousands of miles, with researchers and thinkers from this field's young history and its sprawling frontier. What I found was a field finding its legs, amid exhilarating and

sometimes terrifying progress. A story I thought I knew showed itself to be, by turns, more riveting, harrowing, and hopeful than I had understood.

Machine learning is an ostensibly technical field crashing increasingly on human questions. Our human, social, and civic dilemmas are becoming technical. And our technical dilemmas are becoming human, social, and civic. Our successes and failures alike in getting these systems to do "what we want," it turns out, offer us an unflinching, revelatory mirror.

This is a story in three distinct parts. Part one explores the alignment problem's beachhead: the present-day systems already at odds with our best intentions, and the complexities of trying to make those intentions explicit in systems we feel capable of overseeing. Part two turns the focus to reinforcement learning, as we come to understand systems that not only predict, but act; there are lessons here for understanding evolution, human motivation, and the delicacy of incentives, with implications for business and parenting alike. Part three takes us to the forefront of technical AI safety research, as we tour some of the best ideas currently going for how to align complex autonomous systems with norms and values too subtle or elaborate to specify directly.

For better or worse, the human story in the coming century is likely to be one of building such systems and setting them, one by one, in motion. Like the sorcerer's apprentice, we will find ourselves just one set of agents among many, in a world crowded—as it were—with brooms.

How, exactly, do we intend to teach them?

And what?

PART I

Prophecy

1

REPRESENTATION

In the summer of 1958, a group of reporters are gathered by the Office of Naval Research in Washington, D.C., for a demonstration by a twenty-nine-year-old researcher at the Cornell Aeronautical Laboratory named Frank Rosenblatt. Rosenblatt has built something he calls the "perceptron," and in front of the assembled press corps he shows them what it can do.

Rosenblatt has a deck of flash cards, each of which has a colored square on it, either on the left side of the card or on the right. He pulls one card out of the deck and places it in front of the perceptron's camera. The perceptron takes it in as a black-and-white, 20-by-20-pixel image, and each of those four hundred pixels is turned into a binary number: 0 or 1, dark or light. The four hundred numbers, in turn, are fed into a rudimentary neural network, the kind that McCulloch and Pitts had imagined in the early 1940s. Each of these binary pixel values is multiplied by an individual negative or positive "weight," and then they are all added together. If the total is negative, it will output a −1 (meaning the square is on the left), and if it's positive, it will output a 1 (meaning the square is on the right).

The perceptron's four hundred weights are initially random, and its outputs, as a result, are nonsense. But every time the system guesses "wrong," Rosenblatt "trains" it, by dialing up the weights that were too low and turning down the weights that were too high.

Fifty of these trials later, the machine now *consistently* tells left-side cards and right-side cards apart, including ones he hasn't shown it before.

The demonstration itself is strikingly modest, but it signifies something grander. The machine is, in effect, learning from

experience—what Rosenblatt calls a "self-induced change in the wiring diagram."[1]

McCulloch and Pitts had imagined the neuron as a simple unit of input and output, of logic and arithmetic, and they had shown the enormous power of such rudimentary mechanisms, in great enough numbers and suitably connected. But they had said next to nothing about how exactly the "suitably connected" part was actually meant to be achieved.[2]

"Rosenblatt made a very strong claim, which at first I didn't believe," says MIT's Marvin Minsky, coincidentally a former classmate of Rosenblatt's at the Bronx High School of Science.[3] "He said that if a perceptron was physically capable of being wired up to recognize something, then there would be a procedure for changing its responses so that eventually it would learn to carry out the recognition. Rosenblatt's conjecture turned out to be mathematically correct, in fact. I have a tremendous admiration for Rosenblatt for guessing this theorem, since it is very hard to prove."

The perceptron, simple as it is, forms the blueprint for much of the machine-learning systems we will go on to discuss. It contains a model *architecture:* in this case, a single artificial "neuron" with four hundred inputs, each with its own "weight" multiplier, which are then summed together and turned into an all-or-nothing output. The architecture has a number of adjustable variables, or *parameters:* in this case, the positive or negative multipliers attached to each input. There is a set of *training data:* in this case, a deck of flash cards with one of two types of shapes on them. The model's parameters are tuned using an optimization algorithm, or *training algorithm*.

The basic training procedure for the perceptron, as well as its many contemporary progeny, has a technical-sounding name —"stochastic gradient descent"—but the principle is utterly straightforward. Pick one of the training data at random ("stochastic") and input it to the model. If the output is exactly what you want, do nothing. If there is a difference between what you wanted and what you got, then figure out in which direction ("gradient") to adjust each weight—whether by literal turning of physical knobs or simply the changing of numbers in software—to lower the error for this

particular example. Move each of them a little bit in the appropriate direction ("descent"). Pick a new example at random, and start again. Repeat as many times as necessary.

This is the basic recipe for the field of machine learning—and the humble perceptron will be both an overestimation and an underestimation of what is to come.

"The Navy," reports the *New York Times*, "revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence."[4]

The *New Yorker* writes that the perceptron, "as its name implies, is capable of original thought." "Indeed," they write, "it strikes us as the first serious rival to the human brain ever devised."

Says Rosenblatt to the *New Yorker* reporter, "Our success in developing the perceptron means that for the first time a non-biological object will achieve an organization of its external environment in a meaningful way. That's a safe definition of what the perceptron can do. My colleague disapproves of all the loose talk one hears nowadays about mechanical brains. He prefers to call our machine a self-organizing system, but, between you and me, that's precisely what any brain is."[5]

That same year, *New Scientist* publishes an equally hopeful, and slightly more sober, article called "Machines Which Learn."[6] "When machines are required to perform complicated tasks it would often be useful to incorporate devices whose precise mode of operation is not specified initially," they write, "but which learn from experience how to do what is required. It would then be possible to produce machines to do jobs which have not been fully analysed because of their complexity. It seems likely that learning machines will play a part in such projects as the mechanical translation of languages and the automatic recognition of speech and of visual patterns."

"The use of the term 'learning machine' invites comparison with the learning of people and animals," the article continues. "The drawing of analogies between brains and machines requires caution to say the least, but in a general way it is stimulating for workers in either field to know something of what is happening in the other, and it is possible that speculation about machines which learn may

eventually produce a system which is a true analogue of some form of biological learning."

The history of artificial intelligence is famously one of cycles of alternating hope and gloom, and the Jetsonian future that the perceptron seemed to herald is slow to arrive.

Rosenblatt, with a few years of hindsight, will wish the press had used a bit more caution in their reactions to his invention. The popular press "fell to the task with all of the exuberance and sense of discretion of a pack of happy bloodhounds," he says—while admitting, on his own behalf, a certain "lack of mathematical rigor in preliminary reports."[7]

Minsky, despite his "tremendous admiration" for Rosenblatt and his machine, begins "to worry about what such a machine could not do." In 1969, he and his MIT colleague Seymour Papert publish a book called *Perceptrons* that effectively slams the door shut on the entire vein of research. Minsky and Papert show, with the stiff formality of mathematical proof, that there are seemingly basic patterns that Rosenblatt's model simply will never be able to recognize. For instance, it is impossible to train one of Rosenblatt's machines to recognize when a card has an *odd* versus an *even* number of squares on it. The only way to recognize more complex categories like this is to use a network with multiple layers, with earlier layers creating a *representation* of the raw data, and the later layers operating on the representation. But no one knows how to tune the parameters of the early layers to make representations useful for the later ones. The field hits the equivalent of a brick wall. "There had been several thousand papers published on perceptrons up to 1969," says Minsky.

"Our book put a stop to those."[8]

It is as if a dark cloud has settled over the field, and everything falls apart: the research, the money, the people. Pitts, McCulloch, and Lettvin, who have all three moved to MIT, are sharply exiled after a misunderstanding with MIT's Norbert Wiener, who had been like a second father figure to Pitts and now won't speak to him. Pitts, alcoholic and depressed, throws all of his notes and papers into a fire, including an unpublished dissertation about three-dimensional neural

networks that MIT tries desperately to salvage. Pitts dies from cirrhosis in May 1969, at the age of 46.[9] A few months later Warren McCulloch, at the age of 70, succumbs to a heart seizure after a long series of cardiopulmonary problems. In 1971, while celebrating his 43rd birthday, Frank Rosenblatt drowns in a sailing accident on the Chesapeake Bay.

By 1973, both the US and British governments have pulled their funding support for neural network research, and when a young English psychology student named Geoffrey Hinton declares that he wants to do his doctoral work on neural networks, again and again he is met with the same reply: "Minsky and Papert," he is told, "have proved that these models were no good."[10]

THE STORY OF ALEXNET

It is 2012 in Toronto, and Alex Krizhevsky's bedroom is too hot to sleep. His computer, attached to twin Nvidia GTX 580 GPUs, has been running day and night at its maximum thermal load, its fans pushing out hot exhaust, for two weeks.

"It was very hot," he says. "And it was loud."[11]

He is teaching the machine how to see.

Geoffrey Hinton, Krizhevsky's mentor, is now 64 years old and has not given up. There is reason for hope.

By the 1980s it became understood that networks with multiple layers (so-called "deep" neural networks) *could*, in fact, be trained by examples just as a shallow one could.[12] "I now believe," admitted Minsky, "that the book was overkill."[13]

By the late '80s and early '90s, a former postdoc of Hinton's named Yann LeCun, working at Bell Labs, had trained neural networks to identify handwritten numerals from 0 to 9, and neural networks found their first major commercial use: reading zip codes in post offices, and deposit checks in ATMs.[14] By the 1990s, LeCun's networks were processing 10 to 20% of all checks in the United States.[15]

But the field hit another plateau, and by the 2000s, researchers were still largely stuck fiddling with databases of handwritten zip

codes. It was understood that, in *principle*, a big-enough neural network, with enough training examples and time, can learn almost anything.[16] But no one had fast-enough computers, enough data to train on, or enough patience to make good on that theoretical potential. Many lost interest, and the field of computer vision, along with computational linguistics, largely moved on to other things. As Hinton would later summarize, "Our labeled datasets were thousands of times too small. [And] our computers were millions of times too slow."[17] Both of these things, however, would change.

With the growth of the web, if you wanted not fifty but five hundred thousand "flash cards" for your network, suddenly you had a seemingly bottomless repository of images. There was only one problem, which was that they usually didn't come with their category label readily attached. You couldn't train a network unless you knew what the network's output was supposed to be.

In 2005, Amazon launched its "Mechanical Turk" service, allowing for the recruiting of human labor on a large scale, making it possible to hire thousands of people to perform simple actions for pennies a click. (The service was particularly well suited to the kinds of things that future AI is thought to be able to do—hence its tagline: *artificial* artificial intelligence.) In 2007, Princeton professor Fei-Fei Li used Amazon Mechanical Turk to recruit human labor, at a scale previously unimaginable, to build a dataset that was previously impossible. It took more than two years to build, and had three *million* images, each labeled, by human hands, into more than five thousand categories. Li called it ImageNet, and released it in 2009. The field of computer vision suddenly had a mountain of new data to learn from, and a new grand challenge. Beginning in 2010, teams from around the world began competing to build a system that can reliably look at an image—dust mite, container ship, motor scooter, leopard—and say what it is.

Meanwhile, the relatively steady progress of Moore's law throughout the 2000s meant that computers could do in minutes what the computers of the 1980s took days to do. One further development, however, turned out to be crucial. In the 1990s, the video-game industry began to produce dedicated graphics processors called GPUs,

designed to render complex 3D scenes in real time; instead of executing instructions with perfect precision one after another, as a traditional CPU does, they are capable of doing a great many simple and sometimes approximate calculations at once.[18] Only later, in the mid-2000s, did it come to be appreciated that the GPU could do a lot more than light and texture and shadow.[19] It turned out that this hardware, designed for computer gaming, was in fact tailor-made for training neural networks.

At the University of Toronto, Alex Krizhevsky had taken a class on writing code for GPUs, and decided to try it on neural networks. He applied himself to a popular image-recognition benchmark called CIFAR-10, which contained thumbnail-sized images that each belonged to one of ten categories: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, or truck. Krizhevsky built a network and began using a GPU to train it to categorize CIFAR-10 images. Shockingly, he was able to train his network from a random starting configuration all the way to state-of-the-art accuracy. In *eighty seconds*.[20]

It is at this point Krizhevsky's labmate, Ilya Sutskever, takes notice and offers him what will become a kind of siren song. "I bet," Sutskever says, "you can make it work on ImageNet."

They build an enormous neural network: 650,000 artificial neurons, arranged into 8 layers, connected by 60 million adjustable weights. In his bedroom at his parents' house, Krizhevsky starts showing it pictures.

Step by step, piece by piece, the system gets a few percent more accurate.

The dataset—as big as it is, a few million pictures—isn't enough. But Krizhevsky realizes he can fake it. He starts doing "data augmentation," feeding the network mirror images of the data. That seems to help. He feeds it images that are cropped slightly, or tinted slightly. (A cat, after all, still looks like a cat when you lean forward or to the side, or go from natural to artificial light.) This seems to help.

He plays with different architectures—this number of layers, that number of layers—groping more or less blindly for what configuration might just happen to work best.

Krizhevsky occasionally loses the faith. Sutskever never does. Time and again he spurs Krizhevsky on. *You can make it work.*

"Ilya was like a religious figure," he says. "It's always good to have a religious figure."

Trying out a new version of the model, and training it until the accuracy maxed out, takes about two weeks, running twenty-four hours a day—which means that the project, though at some level frantic, also has a lot of downtime. Krizhevsky thinks. And tinkers. And waits. Hinton has come up with an idea called "dropout," where during training certain portions of the network get randomly turned off. Krizhevsky tries this, and it seems, for various reasons, to help. He tries using neurons with a so-called "rectified linear" output function. This, too, seems to help.

He submits his best model on the ImageNet competition deadline, September 30, and then the final wait begins.

Two days later, Krizhevsky gets an email from Stanford's Jia Deng, who is organizing that year's competition, cc'd to all of the entrants. In plain, unemotional language, Deng says to click the link provided to see the results.

Krizhevsky clicks the link provided and sees the results.

Not only has his team won, but they have *obliterated* the rest of the entire field. The neural network trained in his bedroom—its official name is "SuperVision," but history will remember it simply as "AlexNet"—made *half* as many errors as the model that came in second.

By the Friday of the conference, when it is time for the ImageNet Large Scale Visual Recognition Challenge workshop, the word has spread. Krizhevsky has been given the final talk slot of the day, and at 5:05 p.m. he takes his place up at the presenter's lectern. He looks around the room. In the front row is Fei-Fei Li; to the side is Yann LeCun. There is a majority, it seems, of the leading computer vision researchers in the world. The room is over capacity, with people standing along the aisles and walls.

"I was nervous," he says. "I was not comfortable."

And then, in front of the standing-room audience, not comfortable, Alex Krizhevsky tells them everything.

When Frank Rosenblatt was interviewed about his perceptron in 1958, he was asked what practical or commercial uses a machine like the perceptron might have. "At the moment, none whatever," he replied cheerfully.[21]

"In these matters, you know, use follows invention."

THE PROBLEM

On Sunday evening, June 28, 2015, web developer Jacky Alciné was at home watching the BET Awards when he got a notification that a friend had shared a picture with him through Google Photos. When he opened Google Photos, he noticed the site had been redesigned. "I was like, 'Oh, the UI's changed!' I remembered I/O [Google's annual software developer conference] happened, but I was curious; I clicked through."[22] Google's image recognition software had automatically identified groups of photos, and gave each a thematic caption. "Graduation," said one—and Alciné was impressed that the system had managed to identify the mortarboard and tassel on his younger brother's head. Another caption stopped him cold. The album cover was a selfie of Alciné and a friend of his. Alciné is Haitian-American; both he and his friend are Black.

"Gorillas," it said.

"So I thought—To be honest, I thought that *I* did something." He opened the album, expecting he had somehow misclicked or mistagged something. The album was full of dozens of photos of Alciné and his friend. And nothing else. "I'm like—This was seventy-plus photos. There's no way. . . . That's actually where I really realized what happened."

Alciné took to Twitter. "Google Photos," he wrote, "y'all fucked up. My friend's not a gorilla."[23]

Within two hours, Google+ chief architect Yonatan Zunger reached out. "Holy fuck," he wrote. "This is 100% Not OK."

Zunger's team deployed a change to Google Photos within another few hours, and by the following morning, only two photos were still mislabeled. Then Google took a more drastic step: they removed the label entirely.

In fact, three years later, in 2018, *Wired* reported that the label "gorilla" was *still* manually deactivated on Google Photos. That means that, years later, nothing will be tagged as a gorilla, including gorillas.[24]

Curiously, the press in 2018, just as in 2015, appeared to repeatedly mischaracterize the nature of the mistake. Headlines proclaimed, "Two Years Later, Google Solves 'Racist Algorithm' Problem by Purging 'Gorilla' Label from Image Classifier"; "Google 'Fixed' Its Racist Algorithm by Removing Gorillas from Its Image-Labeling Tech"; and "Google Images 'Racist Algorithm' Has a Fix But It's Not a Great One."[25]

Being himself a programmer and familiar with machine-learning systems, Alciné knew the issue wasn't a biased *algorithm*. (The algorithm was stochastic gradient descent, just about the most generic, vanilla, allpurpose idea in computer science: go through your training data at random, tune your model's parameters to assign slightly higher probability to the correct category for that image, and repeat as needed.) No, what he immediately sensed was that something had gone terribly awry in the training data itself. "I couldn't even blame the algorithm," he says. "It's not even the algorithm at fault. It did exactly what it was designed to do."

The problem, of course, with a system that can, in theory, learn just about anything from a set of examples is that it finds itself, then, at the mercy of the examples from which it's taught.

CALIBRATION AND THE HEGEMONY OF DESIGN

> The extent to which we take everyday objects for granted is the precise extent to which they govern and inform our lives.
>
> — MARGARET VISSER[26]

The single most photographed American of the nineteenth century—more than Abraham Lincoln or Ulysses S. Grant—was Frederick Douglass, the abolitionist author and lecturer who had himself escaped from slavery at the age of twenty.[27] This was no accident; for Douglass, the photograph was just as important as the essay or the

speech. The photograph was just coming into its own through the daguerreotype in the 1840s, and Douglass immediately understood its power.

Before the photograph, representations of Black Americans were limited to drawings, paintings, and engravings. "Negroes can never have impartial portraits at the hands of white artists," Douglass wrote. "It seems to us next to impossible for white men to take likenesses of black men, without most grossly exaggerating their distinctive features."[28] One exaggeration, in particular, prevailed during Douglass's time. "We colored men so often see ourselves described and painted as monkeys, that we think it a great piece of good fortune to find an exception to this general rule."[29]

The photograph not only countered such caricatures but, further, made possible a kind of transcending empathy and recognition. "Whatever may be the prejudices of those who may look upon it," said Douglass of a photograph of the first Black US senator, Hiram Revels, "they will be compelled to admit that the Mississippi Senator is a man."[30]

But all was not entirely well. As photography became more standardized and mass-produced in the twentieth century, some began to feel that the field of photography was itself worthy of critique. As W.E.B. Du Bois wrote in 1923, "Why do not more young colored men and women take up photography as a career? The average white photographer does not know how to deal with colored skins and having neither sense of the delicate beauty or tone nor will to learn, he makes a horrible botch of portraying them."

We often hear about the lack of diversity in film and television—among casts and directors alike—but we don't often consider that this problem exists not only in front of the camera, not only behind the camera, but in many cases *inside* the camera itself. As Concordia University communications professor Lorna Roth notes, "Though the available academic literature is wide-ranging, it is surprising that relatively few of these scholars have focused their research on the skin-tone biases within the actual apparatuses of visual reproduction."[31]

For decades, she writes, film manufacturers and film developers used a test picture as a color-balance benchmark. This test picture became known as the "Shirley card," named after Shirley Page, a Kodak employee and the first model to pose for it.[32] It perhaps goes without saying that Shirley and her successors were overwhelmingly White. The chemical processing of film was tuned accordingly, and as a result cameras simply didn't take good photos of Black people.

(In video just as in photography, colors have for decades been calibrated to White skin. In the 1990s, Roth interviewed one of the camera operators on *Saturday Night Live* about the process of tuning the cameras before broadcast. He explained, "A good VCR person will have a color girl stand in front of the cameras and stay there while the technicians focus on her flesh tones to do their fine adjustments to balance the cameras. This color girl is always white.")[33]

Amazingly, Kodak executives in the 1960s and '70s described the major impetus for making film that was sensitive to a wider range of darker tones as having come not from the civil rights movement but from the *furniture and chocolate industries*, which complained that film wasn't properly showing the grains of darker woods, or the difference between milk and dark chocolate.[34]

Former manager of Kodak Research Studios Earl Kage reflects on this period of research: "My little department became quite fat with chocolate, because what was in the front of the camera was consumed at the end of the shoot." Asked about the fact that this was all happening against the backdrop of the civil rights movement, he adds, "It is fascinating that this has never been said before, because it was never Black flesh that was addressed as a serious problem that I knew of at the time."[35]

In time Kodak began using models of more diverse skin tones. "I started incorporating black models pretty heavily in our testing, and it caught on very quickly," recalls Kodak's Jim Lyon. "I wasn't attempting to be politically correct. I was just trying to give us a chance of making a better film, one that reproduced everybody's skin tone in an appropriate way."

By the 1990s, the official Kodak Shirley card now had three different models on it, of different races. Their Gold Max film—initially marketed with the claim that it could photograph a "dark horse in low light"—now featured in television commercials with diverse families. One depicts a Black boy in a bright white karate gi, smiling as he performs a kata and presumably receives his next belt. It says, "Parents, would you trust this moment to anything other than Kodak Gold film?"

Their original target audience had given them a problematic calibration measure. Now a new calibration measure had given them a new audience.

FIXING THE TRAINING SET

All machine-learning systems, from the perceptron onward, have a kind of Shirley card at their heart: namely, the set of data on which they were trained. If a certain type of data is underrepresented or absent from the training data but present in the real world, then all bets are off.[36]

As UC Berkeley's Moritz Hardt argues, "The whole spiel about big data is that we can build better classifiers largely as a result of having more data. The contrapositive is that less data leads to worse predictions. Unfortunately, it's true by definition that there is always proportionately less data available about minorities. This means that our models about minorities generally tend to be worse than those about the general population."[37]

Alciné's frustrated tweets the night of the incident echo exactly this sentiment. He's a software engineer. He instantly diagnoses what has gone wrong. Google Photos, he infers, just didn't have nearly as many pictures of Black people in it as pictures of White people. And so the model, seeing anything unfamiliar, was much more prone to error.

"Again, I can completely understand how that happens," Alciné tells me.[38] "Like if you take a picture of an apple, but only red apples, when it sees a green apple it might think it's a pear. . . . Little things like that. That I understand. But then, you're the world's—Your

mission is to index the entire world's social knowledge, so how did you, like, just skip over an entire *continent* of people?"

The problems of the twentieth century appear to be repeating themselves uncannily in the twenty-first. Fortunately, it seems that some of the solutions are, too. All it would take was someone willing to question exactly who and what were represented in these twenty-first-century "Shirley cards," anyway—and what a better one might look like.

When Joy Buolamwini was a computer science undergrad at Georgia Tech in the early 2010s, she was given an assignment to program a robot to play peekaboo. The programming part was easy, but there was one issue: the robot wouldn't recognize Buolamwini's face. "I borrowed my roommate's face to get the project done, submitted the assignment, and figured, 'You know what, somebody else will solve this problem.' "[39]

Later in her undergraduate studies, she traveled to Hong Kong for an entrepreneurship competition. A local startup was giving a demo of one of its "social robots." The demo worked on everyone in the tour group . . . except for Buolamwini. As it happened, the startup was using the *very* same off-the-shelf, open-source face-recognition code that she herself had used back at Georgia Tech.

In one of the first articles explicitly addressing the notion of bias in computing systems, the University of Washington's Batya Friedman and Cornell's Helen Nissenbaum had warned that "computer systems, for instance, are comparatively inexpensive to disseminate, and thus, once developed, a biased system has the potential for widespread impact. If the system becomes a standard in the field, the bias becomes pervasive."[40]

Or, as Buolamwini herself puts it, "Halfway around the world, I learned that algorithmic bias can travel as quickly as it takes to download some files off of the internet."[41]

After a Rhodes Scholarship at Oxford, Buolamwini came to the MIT Media Lab, and there she began working on an augmented-reality project she dubbed the "Aspire Mirror." The idea was to project empowering or uplifting visuals onto the user's face—making the onlooker transform into a lion, for instance. Again, there was only

one problem. The Aspire Mirror only worked on Buolamwini herself when she put on a white mask.

The culprit is not stochastic gradient descent; it is, clearly, the sets of images on which these systems are trained. Every face-detection or face-recognition system has, behind it and implicitly within it, a set of images—typically tens or hundreds of thousands—on which the system was originally trained and developed. This training data, the Shirley cards of the twenty-first century, is often invisible, or taken for granted, or absent entirely: a pretrained model disseminated online almost never comes with its training data included. But it is very much present, and will permanently shape the behavior of a deployed system.

A major movement in rooting out bias, then, is trying to better expose, and better understand, the training datasets behind major academic and commercial machine-learning systems.

One of the more popular public-domain databases of pictures of faces, for instance, is what's known as the Labeled Faces in the Wild (LFW) dataset, painstakingly assembled in 2007 from online news articles and image captions by a team from UMass Amherst, and used by innumerable researchers thereafter.[42] The composition of this database was not deeply studied, however, until many years later. In 2014, Michigan State's Hu Han and Anil Jain analyzed the dataset and determined it was more than 77% male, and more than 83% White.[43] The most common individual in the dataset is the person who appeared most often in online news photos in 2007: then-president George W. Bush, with 530 unique images. In fact, there are more than twice as many images of George W. Bush in the LFW dataset as there are of all Black women, combined.[44]

The original 2007 paper describing the database noted that a set of images gathered from online news articles "clearly has its own biases," but these "biases" are considered from a technical, rather than social, standpoint: "For example, there are not many images which occur under extreme lighting conditions, or very low lighting conditions." Such lighting issues aside, the authors write, "the range and diversity of pictures present is very large."

Twelve years later, however, in the fall of 2019, a disclaimer suddenly appeared on the webpage of the Labeled Faces in the Wild dataset that takes a different view. It notes, "Many groups are not well represented in LFW. For example, there are very few children, no babies, very few people over the age of 80, and a relatively small proportion of women. In addition, many ethnicities have very minor representation or none at all."[45]

In recent years, greater attention has been paid to the makeup of these training sets, though much remains to be done. In 2015, the United States Office of the Director of National Intelligence and the Intelligence Advanced Research Projects Activity released a face image dataset called IJB-A, boasting, they claimed, "wider geographic variation of subjects."[46] With Microsoft's Timnit Gebru, Buolamwini did an analysis of the IJB-A and found that it was more than 75% male, and almost 80% light-skinned. Just 4.4% of the dataset were dark-skinned females.[47]

Eventually it became clear to Buolamwini that the "somebody else [who] will solve this problem" was—of course—her. She started a broad investigation into the current state of face-detection systems, which became her MIT thesis. She and Gebru set out first to build a dataset with a more balanced representation of both gender and skin tone. But where would they get their images from? Previous datasets, drawing from online news, for instance, were totally imbalanced. They decided on *parliaments*, compiling a database of the representatives of six nations: Rwanda, Senegal, South Africa, Iceland, Finland, and Sweden. This dataset was notably *un*diverse in things like age, lighting, and pose, with almost all subjects middle-aged or older, centered in the frame, and looking straight into the camera with a neutral or slightly smiling expression. But, measured by skin and by gender, it was arguably the most diverse machine-learning dataset assembled to date.[48]

With this parliamentarian dataset in hand, Buolamwini and Gebru looked at three commercially available face-classification systems—from IBM, Microsoft, and the Chinese firm Megvii, maker of the widely used Face++ software—and ran each through the paces.

Across the dataset as a whole, all three systems did reasonably well at correctly classifying the gender of the subject—approximately 90% for all three companies. In all three cases, the software was roughly 10 to 20% more accurate on male faces than female faces, and all were also roughly 10 to 20% more accurate on lighter faces than darker ones. But when Buolamwini did an intersectional analysis of the two, the starkest result by far appeared. All three systems were *dramatically* worse at classifying faces that were both dark-skinned *and* female. IBM's system, for instance, had an error rate of only 0.3% for light-skinned males, but 34.7% for dark-skinned females: more than a *hundredfold* greater.

The abolitionist and women's rights activist Sojourner Truth is arguably best known for her 1851 speech "Ain't I a Woman?" Buolamwini poignantly echoes this question into the twenty-first century, pointing to photos of Truth that are miscategorized, again and again and again, by contemporary commercial face-classification software, as male.[49]

On December 22, 2017, Buolamwini reached out to the three firms with her results, explaining that she would be presenting them at an upcoming conference, and giving each an opportunity to respond. Megvii did not respond. Microsoft responded with a generic statement: "We believe the fairness of AI technologies is a critical issue for the industry and one that Microsoft takes very seriously. We've already taken steps to improve the accuracy of our facial recognition technology and we're continuing to invest in research to recognize, understand and remove bias."[50] IBM, however, was another story entirely. They responded the same day, thanked Buolamwini for reaching out, replicated and confirmed her results, invited her to both their New York and Cambridge campuses, and within a matter of weeks announced a new version of their API with a *tenfold* improvement in the error rate for dark-skinned females.[51]

"Change is possible," she says. There was no fundamental obstacle, technological or otherwise, to equalizing this performance gap; it just took someone asking the right questions.

Buolamwini and Gebru's work highlights the skepticism we ought to feel when a company announces that their system is, say, "99%

accurate": Accurate on what? Accurate for whom? It's a reminder, too, that every machine-learning system *is* a kind of parliament, in which the training data represent some larger electorate—and, as in any democracy, it's crucial to ensure that everyone gets a vote.[52]

Bias in machine-learning systems is often a direct result of the data on which the systems are trained—making it incredibly important to understand who is represented in those datasets, and to what degree, before using them to train systems that will affect real people.

But what do you do if your dataset is as inclusive as possible—say, something approximating the entirety of written English, some hundred billion words—and it's the world *itself* that's biased?

THE DISTRIBUTIONAL HYPOTHESIS: WORD EMBEDDINGS

> You shall know a word by the company it keeps.
>
> — J. R. FIRTH[53]

Let's say you find a message in a bottle, washed up on a beach; a couple parts of the message are unreadable. You examine one sentence: "I hath buried the treasure north of the ———by the beach." Needless to say, you are highly motivated to figure out what the missing word might be.

It probably does not occur to you that the word might be "hamster" or "donut" or "toupee." There are a few reasons for this. You have some common sense you can apply: hamsters are restless, donuts are biodegradable, and toupees blow in the wind—none of them reliable landmarks for long-term treasure wayfinding. Anyone hiding loot over a span of presumably months to years, you reason, would need something stable, unlikely to disintegrate or move.

Now imagine you're a computer with a complete lack of such common sense—let alone the ability to put yourself in the shoes of a prospective treasure burier—but what you do have is an extremely large sample (a "corpus") of real-world texts to scan for patterns. How

good a job could you do at predicting the missing word *purely* based on the statistics of the language itself?

Constructing these kinds of predictive models has long been a grail for computational linguists.[54] (Indeed, Claude Shannon founded information theory in the 1940s on a mathematical analysis of this very sort, noticing that some missing words are more predictable than others, and attempting to quantify by how much.[55]) Early methods involved what are known as "*n*-grams," which meant simply *counting up* every single chain of, say, two words in a row that appeared in a particular corpus—"appeared in," "in a," "a particular," "particular corpus"—and tallying them in a huge database.[56] Then it was simple enough, given a missing word, to look at the preceding word and find which *n*-gram in the database beginning with that preceding word had appeared most often. That would then be your best guess as to what was missing. Of course, additional context beyond just the immediately preceding word could offer you additional clues, but incorporating it was far from straightforward. Going from storing a list of all possible two-word phrases in your language ("bigrams") to all the three-word phrases ("trigrams"), or fourword phrases or more, meant growing your database to an absurd and untenable size. Moreover, these databases became incredibly sparsely populated, with the vast majority of possible phrases never appearing at all, and much of the rest appearing only once or twice.

Ideally, we'd also want to be able to make reasonable guesses even if a particular phrase had *never* appeared verbatim before in the corpus. Such counting-based methods are no help. In the sentence "I sipped at a jaundiced ———," we might imagine "chardonnay" is more likely than "charcoal," even if neither word has ever been preceded by "jaundiced" in the history of the language. Relying on a count simply won't help in cases like this—and, again, the problem gets worse the more context we try to add, because the longer the phrases we consider, the more likely we are never to have seen something before.

This set of issues is known as the "curse of dimensionality," and has plagued this linguistic approach from the very beginning.[57]

Was there a better way?

There was, and it came in the form of what are called "distributed representations."[58] The idea was to try to represent words by points in some kind of abstract "space," in which related words appear "nearer" to one another. A number of techniques emerged over the 1990s and 2000s for doing this,[59] but one in particular in the past decade has shown exceptional promise: neural networks.[60]

The hypothesis here, the big bet on which the model rests, is simply this: Words will tend to be found near words that are "similar" to themselves. And these similarities can be captured numerically. The neural network model works by transforming ("embedding") every word into a set (a "vector") of numbers that represent its "coordinates" in that space. This set of coordinate numbers is known as the word's *representation*. (In the case of word2vec, it's three hundred decimal numbers between −1.0 and 1.0.) This enables a direct measure of how "similar" any word is to any other: How far away are those coordinates?[61]

All we have to do is—*somehow*—arrange the words in this space to make them do as good a job of predicting these missing words as possible. (At least, we'll have done as good a job as this particular model architecture allows.)

How are we going to arrive at these representations? Why, of course, stochastic gradient descent! We will simply scatter our words *randomly* throughout space to begin with. Then we'll pick a randomly selected phrase from our corpus, hide a word, and ask the system what it expects might fill in that blank.

When our model guesses wrong, we'll adjust the coordinates of our word representations to slightly nudge the correct word *toward* the context words in our mathematical space and slightly nudge any incorrect guesses *away*. After we make this tiny adjustment, we'll pick another phrase at random and go through this process again. And again. And again. And again. And again.[62]

"At this point," explains Stanford computational linguist Christopher Manning, "sort of a miracle occurs."

In his words:

> It's sort of surprising—but true—that you can do no more than set up this kind of prediction objective, make it the job of every word's word vectors to be such that they're good at predicting the words that appear in their context or vice-versa—you just have that very simple goal—and you say *nothing* else about how this is going to be achieved—but you just pray and depend on the magic of deep learning. . . . And this miracle happens. And out come these word vectors that are just amazingly powerful at representing the meaning of words and are useful for all sorts of things.[63]

In fact, one could argue that these embeddings actually manage to capture *too much* of the nuance of our language. Indeed, they capture with startling clarity the parts we ourselves prefer not to see.

THE DARK SIDE OF EMBEDDINGS

> Out of the crooked timber of humanity no truly straight thing was ever made.
>
> — IMMANUEL KANT[64]

Word-embedding models like these, including Google's word2vec and Stanford's GloVe, subsequently became the de facto standard for computational linguistics, undergirding since roughly 2013 almost every application that involves computer use of language, be it ranking search results, translating passages from one language to another, or analyzing consumer sentiment in written reviews.[65]

Indeed, the embeddings, simple as they are—just a row of numbers for each word, based on predicting nearby missing words in a text—seemed to capture a *staggering* amount of real-world information.

You could, for instance, simply add two vectors together to get a new vector, and search for the nearest word. The results, as we have seen, often made a shocking amount of sense:

```
Czech + currency = koruna
Vietnam + capital = Hanoi
```

```
German + airlines = Lufthansa
French + actress = Juliette Binoche*
```

* With second place going to `Vanessa Paradis` and third to `Charlotte Gainsbourg`.

And you could *subtract* words, too. This meant—incredibly—you could produce "analogies" by getting the "difference" between two words and then "adding" it to a third.[66]

These analogies suggested that the embeddings had captured geography:

```
Berlin - Germany + Japan = Tokyo
```

And grammar:

```
bigger - big + cold = colder
```

And cuisine:

```
sushi - Japan + Germany = bratwurst
```

And science:

```
Cu - copper + gold = Au
```

And tech:

```
Windows - Microsoft + Google = Android
```

And sports:

```
Montreal Canadiens - Montreal + Toronto =
Toronto Maple Leafs
```
[67]

Unfortunately, as we've seen, that wasn't all the vectors captured. They contained stunning gender biases. For every clever or apt analogy for `man:woman`, like `fella:babe`, or `prostate`

`cancer:ovarian cancer`, there was a host of others that seemed to be reflecting mere stereotypes, like `carpentry:sewing`, or `architect:interior designer`, or `doctor:nurse`.

We are only now coming to a full appreciation of the problem. "There have been hundreds of papers written about word embeddings and their applications, from Web search to parsing Curricula Vitae," as Tolga Bolukbasi, Adam Kalai, and their collaborators write. "However, none of these papers have recognized how blatantly sexist the embeddings are and hence risk introducing biases of various types into real-world systems."[68]

Machine-learning systems like this not only *demonstrate* bias but may silently, subtly *perpetuate* it. Consider an employer, doing a search for "software engineer" candidates. The search will rank millions of possible résumés by some measure of "relevance" and present just the top handful.[69] A system naïvely using word2vec, or something like it, might well observe that `John` is a word more typical of engineer résumés than `Mary`. And so, all things being equal, a résumé belonging to John will rank higher in "relevance" than an otherwise identical résumé belonging to Mary. Such examples are more than hypothetical. When one of the clients of Mark J. Girouard, an employment attorney, was vetting a résumé-screening tool from a potential vendor, the audit revealed that one of the two most positively weighted factors in the entire model was the name "Jared." The client did not purchase the résumé-screening tool—but presumably others have.[70]

We already know, of course, that job candidates' names exert influence over real human employers. In 2001 and 2002, the economists Marianne Bertrand and Sendhil Mullainathan mailed out nearly five thousand résumés with randomly assigned names designed to sound either White (Emily Walsh, Greg Baker) or African-American (Lakisha Washington, Jamal Jones). They found a stunning 50% gap in the callback rates, despite the résumés themselves being identical.[71]

Word2vec maps proper names to racial and gender axes just like it does with any other words, putting `Sarah - Matthew` on a gender

axis and `Sarah-Kiesha` on a race axis. Given that it puts professions on these axes as well, it's not hard to imagine a system inadvertently using such racial or gender dimensions—in effect, stereotypes—to uprank or downrank candidates for "relevance" to a given job opening. In other words, we have reason to be every bit as concerned if it is a machine sifting through these résumés, and not a person.[72]

The obvious solution in the human case—removing the names—will not work. In 1952, the Boston Symphony Orchestra began holding its auditions with a screen placed between the performer and the judge, and most other orchestras followed suit in the 1970s and '80s. The screen, however, was not enough. The orchestras realized that they also needed to instruct auditioners, before walking out onto the wood floor of the audition hall, to remove their *shoes*.[73]

The problem with machine-learning systems is that they are designed precisely to infer hidden correlations in data. To the degree that, say, men and women tend toward different writing styles in general—subtle differences in diction or syntax—word2vec will find a host of minute and indirect correlations between `software engineer` and *all* phrasing typical of males.[74] It might be as noticeable as `football` listed among their hobbies, rather than `softball`, as subtle as the names of certain universities or hometowns, or as nearly invisible as a slight grammatical preference for one preposition or synonym over another. A system of this nature cannot, in other words, ever be successfully blindfolded. It will always hear the shoes.

In 2018, Reuters reported that Amazon engineers had, starting in 2014, been developing a machine-learning tool to sift through online résumés and rank possible job candidates from one to five stars—just like Amazon products themselves—based on how promising they seemed, and that Amazon recruiters would focus their efforts accordingly.[75] "They literally wanted it to be an engine where I'm going to give you a hundred résumés, it will spit out the top five, and we'll hire those," as one source told reporters. The measure of that star rating? Similarity—using a word representation model—to résumés of previous Amazon hires in the preceding ten years.

By 2015, however, Amazon began noticing problems. Most of those previous engineering hires were men. The model, they realized, was assigning a negative score to the word "women's"—for instance, in describing extracurriculars. They edited the model to remove this bias.

They also noticed, however, that it was assigning a negative score to the names of all-women's colleges. They edited the model to remove this bias.

Still, the model found a way to hear the shoes. Engineers noticed that the model was assigning positive scores to seemingly *all* vocabulary choices—for instance, words like "executed" and "captured"—more prevalent among male résumés than female ones.[76]

By 2017, Amazon had scrapped the project and disbanded the team that made it.[77]

DEBIASING WORD EMBEDDINGS

For Tolga Bolukbasi and Adam Kalai, along with their BU and Microsoft collaborators, the question was, of course, not simply the *discovery* of these biases but what to *do* about them.

One option was to find the axis in this high-dimensional vector "space" that captured the concept of gender and *delete* it. But deleting the gender dimension altogether would mean losing useful analogies like `king:queen` and `aunt:uncle`. So the challenge, as they put it, is "to reduce gender biases in the word embedding while preserving the useful properties of the embedding."[78]

As it happens, even identifying the proper gender "dimension" is hard. You could define it, for instance, as the vector "difference" of `woman - man`. But there's more going on here than just gender—you also have idiomatic uses like "man oh man" and the verb form, as in "all hands, man your battle stations." The team decided to take a number of different word pairs of this type—`woman - man`, but also `she - he`, `gal - guy`, and so forth—and then use a technique called principal component analysis (PCA) to isolate the axis that explained the greatest amount of the difference in these pairs: presumably, gender.[79]

Then their task was to try to determine, for words that differ on this gender dimension, whether that gender difference is appropriate or inappropriate. Let's say `king` and `queen` are appropriately separated by gender, and ditto `father` and `mother`, but maybe we *don't* want to regard—as word2vec by default does—`Home Depot` as the gender-flipped version of `JC Penney`; or `alcoholism` and `eating disorders`; or `pilot` and `flight attendant`.

How, then, to tease apart the problematic from the unproblematic gender associations for not just a handful but for hundreds of thousands of different words? How to know which analogies should be kept, which should be adjusted, and which should be purged entirely?

The team of five computer scientists found themselves doing, in effect, social science. Indeed, part of the project ended up requiring consultation beyond their normal disciplinary lines. "We're a bunch of machine-learning researchers," says Kalai. "I work in a lab that includes a bunch of social scientists, and just from listening to them talk about various issues that come up in sociology and social sciences, we were aware of these possible concerns that the machine-learning algorithms might discriminate, but none of the five of us—we're all five guys—had ever worked or read much about gender bias."

The group, perhaps naïvely, asked the sociologists how they should encode a formal definition of which analogies were acceptable and which were not. The sociologists rapidly disabused them of the idea that a simple formal definition of this kind was possible. "We're thinking, how do we define the best thing?" says Bolukbasi. "They said, 'Sociologists can't define what is good.' As an engineer you want to say, 'Okay, this is the ideal, so this is my target, so I'm just going to make my algorithm until I reach that target.' Because it's involved so much with people and culture and everything, you don't know what's optimal. You can't optimize for something. It's very hard actually in that sense."

The group decided to identify a set of words that they felt were appropriate to consider as gendered in some essential or fundamental way: words like "he" and "she," "brother" and "sister," as well as