

Dorcas Cummings Lecture

P. READ MONTAGUE

Dr. Read Montague presented the Dorcas Cummings lecture entitled “Connecting Mind and Brain in a Computational Age” to friends and neighbors of Cold Spring Harbor Laboratory and Symposium participants on Saturday, June 2, 2018. Dr. Montague is the Vernon Mountcastle Research Professor at Virginia Tech Carilion School of Medicine, Director of the Human Neuroimaging Laboratory and the Computational Psychiatry Unit at the Fralin Biomedical Research Institute VTC, and Honorary Professor at the Wellcome Centre for Human Neuroimaging at University College London.

I’m going to start the talk by breaking one of my two rules of talks that was taught to me by Gerry Edelman, who was one of my two postdoctoral mentors. This quote is from *The Deep Learning Revolution* by my other postdoctoral mentor, Terry Sejnowski, who is, by any measure, a pioneer in computational neuroscience and bringing it from the back roads into a prominent position in neuroscience that it plays today:

It was often said not too long ago that computer vision was not able to compete with the visual abilities of a one-year-old.

That was a defense, as it were, against “We’ve gotten nowhere with computers and computational systems and neural networks.” This is not true anymore. This is why I asked you about the Tesla [before the talk], because I have friends that let their Tesla drive them to work an hour on Highway 81 and basically never touch the wheel. Maybe they’re foolhardy, but they’re still my friends and not dead yet, so something’s working.

That is no longer true and computers can now recognize objects in images about as well as you can, and there are cars on the road that drive themselves more safely than a 16-year-old. Moreover, computers have not been told how to see or drive, but have learned from experience, following a path that nature took millions of years ago. What is fueling these advances is gushers of data. Data are the new oil. Learning algorithms are refineries that extract information from raw data; information can be used to create knowledge; knowledge underlies understanding, and understanding leads to wisdom.

Now, that’s quite the rhetorical flourish from Terry, so I buy that. I’m going to give some examples of this. Neural network approaches to hard problems were always being poooh-pooohed as, “Oh well, they’re not really playing chess the way people play.” That used to be the defense 30 years ago: “Okay, you’ve got this [artificial] mouse. It runs around a maze, and it falls over on its side, and it’s the size of a vacuum cleaner, and it takes every supercomputer

in the world to keep it running for 45 seconds, but you could never do something as hard as chess.”

Now, chess is gone. The game of Go is gone. Every Atari video game is gone. Computers completely crush human beings on these games, trained in really, really simple ways. The ways they are trained are the same ways that your nervous system trains you to learn about objects in the world. They’re trained on something called “reinforcement learning systems,” about as simple as they could be. That’s what we’re going to focus on here. I actually think the “wisdom” bit is a little bit of a far reach for Terry at the end of that, so we’re just going to stick with the information, knowledge, and understanding part. I won’t be able to impart any wisdom to you today.

So what’s common across these creatures here? Some of them are creatures—I didn’t name them, . . . anyway: honey-bee, rodent, zebra finch, one of my four girls. [Then there’s the programs: TD Gammon, Atari games,] AlphaGo. This is the game of Go. For those of you who don’t know anything about Go, it’s a very complicated game played on a 19×19 board. It’s like Othello on steroids. You capture territory, and there are lots of moves. It’s a very complicated game. It has a gigantic state space, something on the order of 10 to the 150 th power. If I round up, there are about 10 to the 80 th particles in the universe, so 10 to the 150 th is really a big number. The branch factor in Go, meaning the average number of moves from any one board position available to you, is on the order of 250 . As far as I can tell, chess is between a 30 and 40 branch factor: for any given board position, you have 30 to 40 legal moves available to you.

These are really hard problems and that’s why before Deep Blue—IBM’s effort at making a chess-playing program—beat Garry Kasparov in 1997, people thought it couldn’t be done. They thought it was just too hard. I don’t think that proves anything particular about Deep Blue; I think it says something very deep about Garry Kasparov. Garry Kasparov played, basically, the history of chess turbocharged in a supercomputer with built-in grandmaster knowledge and that could do 200 million

look-aheads a second. He plays it to basically a tie, and he can have a political opinion and think about the cheese sandwich he's going to eat after the match. Of course, Deep Blue can't do anything like that. It's not adaptive. Human beings are still pretty cool, but they've been crushed by this.

What do they have in common? The answer is they have algorithms in common. They have learning procedures in common. They have software that's in common. The heart of the programs that play backgammon or Breakout—where you hit a little ping-pong ball and it knocks pieces of a wall out—or Go—the heart of that is something called a “valuation function,” and it's just what you think it is. You put a bunch of data in front of a neural network. The neural network rearranges the data and adjusts a valuation function, which, for our purposes, is basically a look-up table; say I'm in board state 3, what are the values of all the boards, all the 250 moves that I could make now? I'm going to assign a number to every one of them—basically, a giant look-up table. Now, that doesn't seem very human-like. It doesn't seem to be the way that you feel like you think, but the way you think and the instruments that build the way you think are not what you would expect at all.

I'm going to tell you two things. I'm going to tell you the sense in which computational approaches to problems, behavioral problems, and even understanding neural circuits, is an alternative way of looking at the data that's been presented at this symposium, and the data are amazing. The techniques available to neurobiologists now really blow my mind. I've been in the field 25 years, and I would never have imagined we'd be where we are now, asking the kind of questions we can ask.

It's not enough just to understand the parts, because biology's got a lot of variability in it. We kind-of want to know what they're doing, and frankly, learning valuation functions, like these programs do here, is so important. Prediction, which is the heart of how these valuation functions are learned, is so important that I think it's probably been rediscovered by evolution over and over again.

It's thought that eyes evolved anywhere from 40 to 50 separate times. Creatures that use vision to survive have developed all kinds of strategies. We think of this as independent evolution of eyes. Biology has discovered eyes over and over again, all kinds of different eyes. Insect eyes are very different than the eyes in your head, for example. I think the same could be said for learning algorithms, that you should expect these learning algorithms not to be implemented in exactly the same way across phyla, but there could be large patterns that you could maybe detect. I'm going to talk about one of those today, and then I'm going to segue to an area that I think is quite forward looking, and it's just now booting up, that we're going to call “computational psychiatry.”

Biology discovered structural and algorithmic motifs is what I'm arguing. The things in your head and the things in these creatures' heads—honeybee, rat, finch, and human—that do the reinforcement learning are neurons that look like this. This [widely spread, highly dense mass (Fig. 3A in Matsuda et al. [2009], *J Neurosci* 29: 444)]

is the arbor of two axons, and here's a dopamine axon coming up from an area in the brainstem of a rat, projecting to the prefrontal cortex or the striatum. That's one neuron hooked to this mass from down in the brainstem. It creates electrical activity. Little impulses run up to this mass. They divide at each of the branchpoints, and they go out, and they communicate whatever it is they're communicating to this bush here. This is gigantic.

This is in the subesophageal ganglion of the honeybee [Fig. 1A in Hammer (1993), *Nature* 366: 59]. This is a neuron that I know and love called VUM_{mx1} [ventral unpaired median neuron of the maxillary neuromere 1]—this is from the work of Randolph Menzel and the late Martin Hammer—and this projects throughout the bee brain, but it directs a certain kind of reward learning. You have analogous systems in the human being coming from the substantia nigra and the ventral tegmental area [Fig. 1 in Arias-Carrión et al. (2010), *Int Arch Med* 3: 24]. These are clusters of dopamine neurons in your brainstem that, like the rat neuron, project through large expanses of the cortical mantle into an area called the basal ganglia. These are the neurons that you lose when you get Parkinson's disease. If you show up in front of a doctor with symptoms of Parkinson's disease, you've probably already lost 70%–75% of those neurons. Now, there's a cell biology/molecular biology/early trauma insult question of, “How is it that they started dying?,” but once they start dying then your brain doesn't know how to value actions and sequences of actions correctly, and I guess I would say that it's both a motor problem and a valuation problem. Other than your pituitary gland, where there's a little projection from your hypothalamus; this is the only source of dopamine to your brain. If you lose these neurons, that's it. And in a sense, life isn't worth living at that point. You act as though there are no rewarding consequences to any of your thoughts or your actions. It's a really important nexus of neurons there.

They're not the only neurons that do stuff like that. There are neurons that have the same sort of general projection patterns that deliver the chemical serotonin. Serotonin and dopamine—if you looked at just the number of prescriptions written in America for treating or perturbing your serotonin system for depression—SSRIs [selective serotonin reuptake inhibitors], for example—or touching your dopamine system, it's probably 70 million people. So if you were going for the two biggies, you might start there, but there are others, too: acetylcholine, various peptides, et cetera.

This is the bird brain: a zebra finch. This is a cartoon, obviously [Fig. 1B in Gadagkar et al. (2016), *Science* 354: 1278]. It also has something analogous to the ventral tegmental area. This is a dopamine projection, and it controls the way the bird learns to mimic the song of a male.

There's a theme here I'm going to outline in just a second. This is a diagram from a book on computational neuroscience from 1992 when it was barely an embryo—I think I was in utero—*The Computational Brain* written in '92 by Pat Churchland and Terry Sejnowski. I was in the lab at the time at the Salk Institute. This is the classic “scales” diagram [ranging from a 1-m scale for the entire

central nervous system, down through systems and maps at the 10- and 1-cm scales, respectively, past networks at the mm scale, and all the way down to neurons (100 μm), synapses (1 μm), and molecules (1 \AA). This is the classic way that people decompose the problem of understanding the nervous system: scales of space and time. Things at different scales take place at different timescales, spatial scales. They require different kinds of forces, et cetera.

In 1988, they published a paper right before the book, where they plotted “space”—this is eight orders of magnitude, from the size of the brain (1.0–0.1 m) down to a synaptic connection between the two neurons (1.0–0.1 μm)—and these are timescales here from milliseconds up to months. What’s plotted here are the various techniques we have in 2014 for studying this space–time decomposition of the nervous system [Fig. 1 in Sejnowski et al. (2014), *Nat Neurosci* 17: 1440]. It’s a reasonable way to take on the problem. It’s a really hard problem—you hear that in everybody’s talk today, despite the fantastic techniques. But in 1988, this inset in the figure here is what it looked like. One big gap got filled by functional magnetic resonance imaging, and another gap was filled by optogenetics, which actually continues to expand the scales at which it’s been active. I would say that optogenetics probably goes up to the size of the whole brain now, given what I’ve seen in the zebrafish. The techniques are really explosive now. That’s not the only way to decompose a problem though. Another way is to take computational primitives.

I’m going to talk to you about prediction learning. Prediction learning is just what you think it is. In a sense, biology is all about prediction. A creature is challenged by an environmental challenge. Something inside the creature rearranges, and then after that it’s better at responding to a similar or generalized challenge: learning. Bacteria do that. Birds do that. People do that. Herds of people do that. People interacting with machines do that. It’s a principle that you could go looking for at many scales. We’re going to use that to go looking for physical substrates of prediction learning. In particular, we’re interested in how you learn about things that are rewarding to you. How you learn about food, water, sex, and salt: things that keep you alive, and things that keep the species propagating. If you don’t do that, it doesn’t matter whether you drive a Tesla.

Neurons and synapses live in this [0.1–50- μm] range, so we’re going to ask the question, “How can synaptic plasticity—changes in synaptic strengths between neurons—support the kind of learning?” I’m going to dial back to a big idea had by a now very famous psychologist, Donald Hebb. About the same time Jerzy Konorski—a Polish neuroscientist who was somehow overlooked for various political reasons—published the same idea the year before; this is in 1949. Hebb published the following idea based on the following instinct: Correlations and associations in the world seem to change what goes on in your head.

You see the red bird flying overhead, and the next thing you know you see the poop on your picnic table. Every time the red bird comes over, he poops on your table.

There’s something wrong with this red bird. He likes to poop on the table. Eventually, if you’re learning at the right rate, you learn: the red bird comes, I gotta go get the sponge. I’m going to have to clean the poop off the table. Those associations, those correlations, must be changing something in your brain, so Hebb upgraded that to a physiological hypothesis. He said that, suppose that things that look like correlations in the world end up being increased synaptic connectivity between two neurons that correlate with one another. Specifically:

When an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A’s efficiency, as one of the cells firing B, is increased.

This is the “Cells that fire together, wire together” idea. So the bird and its redness and the poop are all happening at the same time. There are neurons that are active during that. They’re coactive, and that coactivity—action potentials or electrical spikes in neuron A paired with electrical activity in neuron B—causes this connection to change its efficiency. Those changes of efficiency are like the numbers that I was talking about in the valuation function for those games earlier. They learn those associations.

Good idea. It was brilliantly simple, and it made lots of sense. And it’s wrong.

This is a paper in 1984 from Eric Kandel, Tom Abrams, Bob Hawkins, and Tom Carew [Carew et al. (1984), *J Neurosci* 4: 1217]. It was a famous paper for me because it came out the year after I graduated from college, and it was testing just that idea. It was testing it on a sea snail that had a little gill withdrawal reflex that it protected itself by. And that gill withdrawal reflex is: You stab the sea snail’s tail, it sucks its gill in, and then it puts it back out and sucks it in. That reflex could be sensitized—made more sensitive and quick—or it could be habituated. They were interested in using this to see if they could look for the cellular basis of that simple learning in that simple creature. And the answer is:

... We have directly tested Hebb’s postulate in *Aplysia* at identified synapses which are known to exhibit a temporally specific increase in efficacy during a cellular analogue of differential conditioning. We find that the mechanism postulated by Hebb is neither necessary nor sufficient to produce the associative change in synaptic strength that underlies conditioning in *Aplysia*.

They noticed that: “In contrast, impulse activity in the presynaptic cell...”—the input end of a synapse—“... must be paired with facilitatory input.” That stuck in my head. I showed up at the Salk Institute in 1990 as a postdoc, as a kind of converted applied mathematician, having gone through the rigors of a neurobiology course at Woods Hole. Terry Sejnowski was the head of the lab. It was an oddball lab in the sense that Howard Hughes supported it back in 1989 or something like that. I met this guy, Peter Dayan. We were littermates in his little hovel. We were assigned a problem: to study diffuse ascending systems and figure out what they do. What information do diffuse ascending systems encode? We were just talking about

these earlier. These are the systems of neurons with their cell bodies in the brainstem that send projections throughout the brain. What are they doing? Why is there such an elaboration of that? What is it about? What kinds of synaptic changes could they direct? That's what we went after first.

The first thing we did was we decided that we needed to change Hebb's Rule. The reason we decided that was a long series of research, really starting in the 1950s with a guy called [Arthur] Samuel who made an automatic checkers playing program and came up with a learning procedure that was super-simple—not quite right, just a hair off what he could have guessed but didn't quite—called “the bucket brigade.” It played checkers, and it could beat humans at checkers; Checkers is a pretty simple game.

There was another guy, Richard Bellman—I think he was at the RAND Corporation at the time—who was working on something called “dynamic programming.” He named it that because he thought it was a happy name. He couldn't see any negative connotations to dynamic programming. Dynamic programming takes on the following problem: You have a game you are playing, so you make a choice, and then you have a set of other choices, and you make one of those and it keeps branching. And you run all the way out to the horizon of the game to whether you lose or win, and it's a principle way to back that process up and learn optimally how to play that game, how to tie together sequences of actions that lead to the most wins. The problem with dynamic programming is it's kind of impossible to do on anything but trivial problems in its native form. Most games, even simple games, are too hard to learn.

Rich Sutton and Andy Barto through the '80s came up with an idea that ended up being something called the “Method of Temporal Differences,” and I'm going to show you that in a second, mainly through waving my hands. A guy called Christopher Watkins at Cambridge University wrote a thesis in 1989 that is one of these singular breakthrough theses where he fused these two ideas. He was working with Peter Dayan—who had just come to the Salk with me—on making a proof of the convergence to this method he called “Q-learning.” It blended together the best aspects of dynamic programming and the best aspects of animal learning. Rich Sutton had based his method on the psychology of animal learning. Even though he ended up with a PhD in computer science, he had been a psychology undergrad at Stanford and had all these ideas about animal learning in his head. In one sentence, Q-learning is a method to learn to maximize your returns over sequences of actions that are possibly long-term.

Let me give you one example. Suppose I took a table and I set it up here, and I put a little cage in the middle of the table. I put a bug in the cage. The cage has a little door in it, but the door is closed. The bug just wanders around inside the cage. Now I open the door and the bug wanders off and it wanders along the table, and it falls off the table and it bumps its head. What made it bump its head? Well, it should assign a lot of negative credit to that last step off that big cliff. You see a big cliff? Stop. Turn around. But the key point was the moment he stepped through the

door. The moment he stepped through the door, every path led to Rome. Every path led to a negative outcome. What Q-learning does incrementally is say, “How do I assign credit to that [early] point for this outcome that's temporally distal?” That's called a temporal credit assignment problem.

So I met them, and they taught me what they were doing with Q-learning, and we all came up with the idea that that was a perfect way to sidestep Eric Kandel's problem from that 1984 paper. He's got something on a presynaptic terminal; it doesn't look like Hebb's Rule, but there's something that depends on it, so we changed Hebb's Rule [Montague et al. (1993), *Adv Neural Inf Process Syst 5*: 969]: “We present a local learning rule in which Hebbian learning...”—correlational learning, correlations in the world translating into correlations in your head storing a parameter—“...is conditional on an incorrect prediction of a reinforcement signal.”—a prediction error signal. Some system is making a prediction of how much reward you're going to get and computing differences between the actual reward you get and the prediction, and that's feeding into synapses. The idea is that action potentials from here [Cell A] correlating with action potentials here [in Cell B] need this other input, and this input controls the sign of the synaptic change. That's small, but it makes a big difference. Now, instead of storing correlations in your head that are consistent out in the world, you store predictions, and you can use a prediction to decide what to do next. That's the idea.

So, here's the way it would look for a mouse running in a maze [T-maze]. Let's discretize the maze: As the mouse progresses, each successive point represents a “state” S , and each point has an associated reward r . So, at the start of the maze, time t , you've got one point, s_t and r_t . Next point, s_{t+1} and r_{t+1} . Next point, s_{t+2} and r_{t+2} . You go right, you get a rock; you go left, you get food. What this algorithm [$V(s_t) = E\{r_t + \gamma \cdot r_{t+1} + \gamma^2 \cdot r_{t+2} + \dots\}$] does—and I'm not going to explain the equation—is it has a goal of learning, and the goal of learning is to learn the “value” of the state at time t : $V(S_t)$. And the value of the state at time t is going to be the average reward you can expect from that state to the future, and we're not going to care how you got in that state. We don't care how you got in any state; it's going to be a history-independent process. But that's the goal of learning: to estimate the value of some board state in Go or some board state in chess, which is going to be the expected value of the rewards you can expect in the future, and you can tie those rewards to either winning the game or losing the game.

When you write down that equation that way, when you say that's the goal of learning, it has a natural error term that falls out of it: [$\delta_t = E\{r_t\} + \gamma \cdot V(s_{t+1}) - V(s_t)$]. This is what we thought the dopamine signal was. And what is this error term? This is called the “temporal difference error term,” and it's a difference in time between the value of the next state you're about to engender—the value of the next state in time—and the value of the current state plus anything unexpected that happens to you. It's a successive prediction model. It goes like this: I predict I'm going to get three units of reward next time. That predic-

tion, plus whether or not I get three units of reward, can change my error term. So, two things can change the error. Your predictions change through time or the reward received is different than what you expected. Both of those things can cause the signal to go high—things were better than expected—or low—things were worse than expected. There are neurons in your brain that act precisely like that, and you can show algorithms that use this sort of setup to learn to play things that can beat every player who's played the game of Go in the world.

So, you have to know a little bit about the history of this Go game. They tuned the AlphaGo game initially using grandmaster input, and they tuned it so that they could play the Chinese champion and the South Korean champion. It won, and it kind of devastated the Go world. Then they made another program called AlphaGo Zero, which trained on no human input at all, and it trains up to criterion in about 10 h and trains to supermaster level in about 30 h, and it beats AlphaGo 100 to zero. It basically beats the entire history of Go.

The complaint now is, "Well, people don't really learn like that. They have these intuitive algorithms and stuff." Maybe, or maybe hard problems like that are just like look-up. You have neurons that emit signals exactly analogous to the signals they used to criticize the moves in those games and learn that value function. I think one way of thinking about that is that biology stumbled into this algorithm multiple times, and I showed you what it did. It ends up with structural and computational motifs that seem to be very analogous when you look at bee brains and mammalian brains and the brains that play Go. Take-home point number 1: We have in our head substrates of prediction and reward that rival the way neural networks now beat what used to be thought of as fantastically difficult problems.

That kind of activity shows up in the activity of dopamine neurons in your brain. Let me just point out something here: This is "time" on the x -axis, and this is average firing rate on the y -axis. Each one of these traces is a single trial. We're listening to a dopamine neuron while an animal is doing a classical conditioning task. There's a cue that comes on: a light. It goes off, you wait a while, then squirt juice in the animal's mouth. When you squirt juice in a naive animal's mouth, the dopamine neurons [fire]—they pop along. These are little electrical impulses in the neurons. If you keep doing that, pairing a cue followed by a reward and you keep the time consistent, all of a sudden when you turn the light on the neurons fire at the cue. They give a burst at the early predicting light. The light predicts the juice, right? And now when the juice is delivered, nothing happens. So, two things have happened. The response to the juice has gone away, and the response to the early light has grown. This is exactly like a temporal difference prediction error, but what's going on silently is, from the time of the cue all the way out to the time of the reward, you've learned a value function. You've learned a set of numbers that says how much juice you can and cannot expect to get, how much reward you can expect to get through those intervening times. That's a simple little algorithm, but it's very powerful when you hook it to a

body and you put a bunch of other constraints and systems in there.

That's a recording from a dopamine neuron. That gets accounted for by the model I've been talking to you about. It's a causal mechanism. It only goes one way through time. It doesn't absorb information if the reward precedes the cued things that follow it. It only goes one way through time. You have to really work hard to make systems learn backward in time. So, causality is built into biology in terms of learning in a very profound and deep way. These are based on the empirical data of Wolfram Schultz [Schultz et al. (1997), *Science* 275: 1593], but it wasn't until 2005 that proper quantitative tests of this got done in a beautiful paper by Hannah Bayer and Paul Glimcher [Bayer and Glimcher (2005), *Neuron* 47: 129]. This paper made me feel really good, that I wasn't wasting my time, and we could move on to other aspects of it.

Just to indicate, bees have systems like this. This is a little neural network model in a bee. Several papers we did on that connect this to decision making [Montague et al. (1993), *Adv Neural Inf Process Syst* 6: 598; Montague et al. (1995), *Nature* 377: 725]. Birdsong learning: same sort of thing [Doya and Sejnowski (1994), *Adv Neural Inf Process Syst* 7: 101]. It has dopamine neurons that teach a bird to copy a template laid down by the male bird that's near it. They listen to it. They use that as an internal goal, and they train up their vocalizations after listening to a male bird for a year. Bees have it. Birds have it. People have it. It's a theme.

When you go test it in human beings, you see the typical pattern. There's a sensory cue that comes on. There's a reward, literally squirting juice in people now [Fig. 1B in McClure et al. (2003), *Neuron* 38: 339]. Using functional magnetic resonance imaging, you can put people in a scanner and you can do that monkey conditioning experiment on them. Initially you see responses early on to the reward. After training, these disappear and they move back to the predictive sensory cue [Fig. 1 in Braver and Brown (2003), *Neuron* 38: 150]. Now, this kind of imaging won't tell you that it's dopamine. It will say that metabolic demand blood flow changes in these regions are following this scheme. These are the people that did these first experiments: Greg Berns, Sam McClure, and a separate group in London led by John O'Doherty [O'Doherty et al. (2003), *Neuron* 38: 329].

You can even go and make recordings of dopamine directly in human brains. This is a group of people that I've worked with over the last 8 years: Ken Kishida, Rosalyn Moran, Paul Phillips. This is people being implanted with deep brain stimulating electrodes for either Parkinson's disease or central tremors. We designed a new way to extract dopamine and serotonin measurements off a small carbon fiber; it's about 7 microns at the tip. You can drop this down into a human brain and have somebody play a betting game. You're endowed with \$100. You make a bet. Market fluctuates up, you win the relative fractional change in the market times your bet. It fluctuates down, you lose the relative fractional change in the market times your bet. When you do that you can see that dopamine produces prediction errors, and it actually tracks the market quite

well. Here's a market trace in blue. That's the value of the market, and these are slow changes in dopamine. The fast changes I was showing you earlier also are seen, but this slow change shows that average dopamine levels in this person's brain were following the market. Now the market crashes. The person loses tons of money, but the dopamine signal pulled out. The person playing the game lost 20% of his initial stake—he ended up with \$80—but the dopamine signal acted like it knew how to track the market and acted like it knew how to pull out of the market.

So we made a little artificial agent network that used the dopamine signal from this person's brain, and we asked, "If you were using that dopamine signal, what kind of decisions could you make?" We let it do the betting. So we took the dopamine signal from this human brain, and we bet "all in" when the last 5 seconds of dopamine was sloping up, and we went "all out" when dopamine signal was going down, and we showed that we made 75% over our \$100. They ended up with \$175. The network ended up with more money than, at the time, the 350 people that had played the game. That's an interesting finding. By doing it in a human it's an interesting finding that this valuation signal in their head is somehow dissociated from their ability to get it all the way out to their finger. That immediately starts saying, "Well, maybe the valuation's intact, and there's something wrong with coupling to movement." You might start sorting patients on that. What we're hoping to do is move this method from this specialized carbon fiber to the standard electrodes that they use and implant in the brain, so that we can make continual measures of this and really find out how we might intervene, and see what the drugs and the stimulation paradigms do.

So what, right? Well, these are all supposed to be "rewarding" things: drugs, chocolate, Krispy Kreme. Why are these things valuable to you? Well, they're valuable to you because systems assigned value to the learning. So let's do a thought experiment. It's a little diabolical; it's a little sick, actually. Let's imagine that on random nights I sneak into your bedroom. I inject you with heroin. I use a really tiny needle. You can't feel it; there's no evidence of it the next day. There's a lot of mosquitoes out. It's summer. I'm really, really clever, and I put it on a random time schedule. Two months later, I'll stop doing it. Are you addicted to heroin? Your body's physiologically dependent on heroin. You feel like you-know-what. You're showing up at the doctor. How are you going to feel? You're going to have diarrhea and a headache, and your body is withdrawing from heroin. But are you addicted to heroin? And the answer is you are not addicted to heroin because there are no cues. I've arranged it so that there are no cues that your nervous system has learned on that you can use to commandeer your behavior and make you reorganize the way you behave. These systems and the way in which these systems are biologically implemented and the way the biological implementations go wrong in terms of the computations are central to understanding drug addiction.

Neuromodulatory systems are central to almost every issue that you could name in psychiatry. Not so many years ago now, I guess 2006, we formed a unit in Houston,

Texas—I used to be at Baylor College of Medicine and Rice University in Houston for 17 years—called the Computational Psychiatry Unit built just on that instinct I gave you there with that little sick illustration. And the idea was, yes, you can decompose the nervous system in all these scales, and we have to know that because that's the biology of the nervous system and those are our knobs that we have to intervene or to understand or to listen in, but you could also alternatively organize that data under computational ideas. And that's what one would do here.

And what did I go to? This is DSM-IV criteria for autism spectrum disorder. I didn't put it up for you to read, but if you peruse any of it, if you know anyone with autism or you have a family member who has autism or is anywhere on the developmental disorder spectrum, you'll know this is a pretty good description. It would fit your experience watching a kid afflicted with autism. What it isn't is it's not a bunch of scientific dimensions. What you need in scientific dimensions are other ways in, and I think one of the things you could have to augment what we do in the area of psychiatry and the treatment of mental illness in terms of the nervous system is a kind of computational psychiatry. So there are units popping up all over the world. There's one in London. There's one being formed in Asia. The National Institute of Mental Health has a funding initiative along these lines.

Let me just give you one example. I went from squirting juice in people's mouths in functional magnetic resonance imaging machines—machines that can eavesdrop safely on your brain activity—to running social exchange tasks with two people at a time. I borrowed games from the game theory world. I crafted them for my own end. I set two people in an interaction. This particular game is called a trust game. One of the things that these games require that you do is think about the other person. What is that other person thinking, and given that they think that, what are they likely to do next?

The trust game goes like this: This guy in red is called the Investor or the Proposer. They're endowed with \$20. They can keep it and the round's over or they can send any fraction of that money to the partner in blue—the Trustee—and it earns a 300% return on the way over. So if I send \$10, it triples to \$30. Control passes to blue, and blue can keep it all or send me any fraction back. Suppose he sends \$15 back, well, I've made \$5 on a risk of \$10. Well, so what? Well, the "so what" is you can form a normative player here. You can make a mathematical rendering of the optimal way to play this game. You can then set real people into interaction in this game, and you can use that mathematical model to decide whether or not they're playing optimally and what-not, and it's quite a good probe. It's a good probe for two reasons, and I've listed them here. One is this same little game engages the same prediction systems that I was talking about earlier for just deciding whether or not this cue predicts you're going to get a drug or juice in your mouth. The same systems are engaged. They shift around according to this model I've shown you, and that's interesting.

But also, they behaviorally parse traditionally defined psychopathology groups. By that I mean you can take a game like this, apply it to a group of people, say 1000 people. Some of them have some psychopathology like borderline personality disorder, major depression, autism spectrum disorder, but a particular case would be autism, where one of the problems is a perturbed capacity to think about the other person, to put yourselves in their shoes. Autism is a pretty large catch-bag so I don't just want to summarize it glibly like that, but any perturbation in your ability to think about the other person and to think about that person's model of you perturbs this interaction. If you can't do those two things, you can't get a job. A job interview is basically image management. I want a job from you, and so I have to have a model of who you think I am and this is who you think I am, but I want it over here, so I have to be able to send signals to you to move my estimate of your model over here. We call that "second-order thought." I can look at you and see how you act. I can model that, and then I have to have some idea of your model of me. Otherwise you can't do interactions like this. That's why we think this is a good game.

I then had this extremely productive postdoc, Brooks King-Casas, who came to the lab and produced just an enormous amount of papers and results and what-not. I'm just going to summarize them quickly. The first thing he showed is you use a game like that, you do functional magnetic resonance imaging, and you see the sorts of shifts I was just showing you earlier. This is a plot of time versus a response of this region of your brain here. This is called the caudate nucleus in your brain. This is where we were recording in human subjects before, during the neurosurgery. If we keep doing the game, the response transfers back to this early point here [Fig. 4 in King-Casas et al. (2005), *Science* 308: 78]. It's not transferring back on stimuli that are going on, it's transferring back based on the plans of the person into the future. In other words, we sorted these signals according to what move this person's going to make 22 seconds into the future. So as they plan their actions into the future, you're engaging what looks like these same prediction error responses in your striatum that would be engaged if you were shining a light and following it with juice, the same systems that you lose in Parkinson's, the same systems that get addicted by every drug of abuse [Fig. 2 in Tomlin et al. (2006), *Science* 312: 1047]. In addition to that, he showed that there were neural responses that you might use to act as a biomarker for both borderline personality disorder [Fig. 1 in King-Casas et al. (2008), *Science* 321: 806] and a separate one that you might use in an assay of autism spectrum disorder.

I then acquired a principal research fellowship from the Wellcome Trust to go after this in a big project between London and Roanoke, Virginia. I set up 19 sites in North-Central London. That took a couple of years, and we've been doing a variety of forms of these two-party exchanges where we record from one or both of the brains, and we're developing neural signatures in the context of these short games of ten rounds; that's all it is.

The instinct here is that healthy humans are sensitive detectors of interpersonal exchange. If there's something

wrong with the way I'm interacting—even if I'm just sending you some money back and forth—and you're not quite fair, or you're wiggling around too much, or you get angry too quickly and the cooperation breaks down, you have a lot of hardware and a lot of software devoted to that. We thought we could use healthy humans as kind of sensitive detectors of what was wrong with the people on the other side of the interaction. We used that instinct and developed a Bayesian method to classify the trajectories through this investment/repayment schedule. So, that's investment one, repayment one... investment two, et cetera, for ten rounds. Using that, we could separate different groups, cluster them easily. We could tell whether or not there was a borderline personality disorder patient on the other side of the interaction. We could tell whether there was a person with major depression on the other side of the interaction. We could tell whether there was somebody with autism on the other side of the interaction [Fig. 2 in Koshelev et al. (2010), *PLoS Comput Biol* 6: e1000966]. Then we can start make computational agents that play in the style of those players and then characterizing those further mathematically [Fig. 6 in Koshelev et al. (2010), *PLoS Comput Biol* 6: e1000966].

That was quite inspiring. It was inspiring to Pearl Chiu. Pearl and Brooks are both professors now, but Pearl was a postdoc from Harvard who was very interested in that. She developed a signature that this game engenders in the brain [Fig. 6 in Chiu et al. (2008), *Neuron* 57: 463]: a brain response that's uniquely elicited by this two-party exchange that is parametrically diminished to the degree to which you have autism. It's a response in something called the cingulate cortex. This is the response of the cingulate cortex on the *x*-axis and this is the ADI [Autism Diagnostic Interview] rating, a caretaker rating of severity of autism. The more this response is gone, the more autistic you are according to that scale. We thought it was related to their model of themselves, and we spent 2 years developing that idea using picture assays where we showed kids pictures of themselves or pictures of other people, pictures of familiar objects, pictures of unfamiliar objects, pictures of favorite objects, and pictures of neutral objects. We developed the idea that we could literally show them one picture of themselves, do functional MRI for 18 seconds, and use that brain response to classify the degree to which they had autism spectrum disorder. That work was by James Lu [Lu et al. (2015), *Clin Psychol Sci* 3: 422]. He was an MD/PhD student at Baylor College of Medicine. It doesn't matter the details of what the response looked like, and it remains to be seen whether or not this will work out, but this is the kind of thing that this two-party exchange engendered. So instead of a blood test, you might be able to take a kind of game test, establish a number, and that could be another arrow in your quiver for thinking about the disease state that they have.

I'm going to end on Breakout. Breakout can't speak to you, but can beat every human on this game where it has to move the paddle around. It was trained on pixel data up to superhuman performance, and what I wanted to point

out was the value function that I use is exactly the thing I showed you before. It's exactly the thing that's in the bee's head. It's exactly the thing that we think is in at least a subclass of dopamine neurons. This is from Google, a company called DeepMind in London [Mnih et al. (2015), *Nature* 518: 529]. It's been wiping out the

board-game-playing world, and now they're trying to develop agents that develop senses of themselves. They're familiar with some of this work that we've done, and I know for a fact they're looking for agent-specific programs that they can use to move into the medical domain.

So, thanks for having me tonight.