EPISODE 383

[INTRODUCTION]

[0:00:00.4] JM: Self-driving cars are here. Fully autonomous systems like Waymo are being piloted in less complex circumstances. Human in the loop systems like Tesla Autopilot navigate drivers when it is safe to do so and they let humans take control in ambiguous circumstances. Computers are great at memorization but not yet great at reasoning. We cannot enumerate to a computer every single circumstance that a car might find itself in. The computer needs to perceive its surroundings, plan how to take actions, execute control the situation and response to changing circumstances inside and outside of the car.

Lex Friedman has worked on autonomous vehicles with companies like Google and Tesla. He recently taught a class on deep learning for semiautonomous vehicles at MIT which is freely available online, on YouTube. I recommend checking out those videos. They are awesome and really informative and actually not too technically dense. It's pretty interesting. Well, there are typically dense parts of it but parts of it are just entertaining.

There was so much ground to cover in this conversation because I have simply not talked in detail to many people who were involved in self-driving cars. Most of the conversation was higher-level. How to even approach this problem? What is the hardware and software architecture of a car? Lex had really satisfactory answers for these. They were not necessarily conclusive, but they were interesting. I really enjoyed talking to Lex, and if you want to hear more from him, check out his podcast. It's called Take It Uneasy, which is about jujitsu. It's about judo, and wrestling, and learning, and self-discipline. Totally unrelated to software engineering, but you might find it interesting especially if you like this episode.

[SPONSOR MESSAGE]

[0:02:07.1] JM: Spring is a season of growth and change. Have you been thinking you'd be happier at a new job? If you're dreaming about a new job and have been waiting for the right time to make a move, go to hire.com/sedaily today. Hired makes finding work enjoyable. Hired uses an algorithmic job-matching tool in combination with a talent advocate who will walk you

through the process of finding a better job. Maybe you want more flexible hours, or more money, or remote work.

Maybe you work at Zillow, or Squarespace, or Postmates, or some of the other top technology companies that are desperately looking for engineers on Hired. You and your skills are in high demand. You listen to a software engineering podcast in your spare time, so you're clearly passionate about technology. Check out hired.com/sedaily to get a special offer for Software Engineering Daily listeners. A $600 signing bonus from Hired when you find that great job that gives you the respect and the salary that you deserve as a talented engineer. I love Hired because it puts you in charge.

Go to hired.com/sedaily, and thanks to Hired for being a continued long-running sponsor of Software Engineering Daily.

[INTERVIEW]

[0:03:36.5] LF: Zencastr doing the — My side recording too?

[0:03:40.5] JM: Yes. Zencastr records a client side. It will throw it in your browser cache and it will upload it at the end. Actually, it streams it to Dropbox and it will upload a WAV file at the end and then you can send me your client side recording also at the end so just in case there's any corruption that occurs in the Zencastr recording I'll have a backup.

[0:04:01.6] LF: Nice. Is that how it happened?

[0:04:03.5] JM: It does happen sometimes. Very rarely. I would say 5% of cases maybe. I guess that's significant, actually.

[0:04:11.0] LF: That is. Actually, I'm asking because I used to do a lot of interviews for different kind of thing, athletes, Olympic athletes and so on. The podcast, I guess, but I haven't talked to —

[0:04:22.2] JM: Due! That's cool.

[0:04:25.2] LF: I prefer sort of the because I can look of these bad-asses in the eye, and it's a little more sort of — The kind of the questions I usually talk about is more about life and fear and overcoming difficult times, so you kind of want to be in-person for some of those.

[0:04:47.3] JM: Listen. Hey, it helps to look somebody in the eye and ask them about reinforcement learning too.

[0:04:52.9] LF: That's true. I agree

[0:04:54.9] JM: What's the name of your Olympic athlete podcast?

[0:04:58.3] LF: It's called — I'm not sure I want to advertise it, but it's Take It Uneasy. Take it easy, but Take It Uneasy.

[0:05:06.1] JM: I love the name. That's great. Can I put that in the show notes? In fact, can I put this little preamble conversation in this show? This is good. I like it.

[0:05:14.8] LF: Yeah, absolutely.

[0:05:16.0] JM: Okay. Cool. All right, I've got a lot of questions to ask you about deep learning and self-driving if you're ready to get into it.

[0:05:22.2] LF: Yeah, let's do it.

[0:05:24.4] JM: Okay. Lex Friedman is a postdoc at MIT. He works on self-driving deep learning for semiautonomous vehicles. Lex, welcome to Software Engineering Daily.

[0:05:35.0] LF: It's good to be here. Thanks, Jeff.

[0:05:37.2] JM: You taught a class in deep learning for semiautonomous vehicles, and early on in that class you ask the question — I think this question recurs throughout your class is, "Is

driving closer to chess or to every day conversation when you're looking at it from the point of view of can we automate this?" Why is that important question to ask?

[0:06:02.9] LF: It's a fascinating question. It's a subset of all kinds of questions that I can summarize as what is intelligence. Trying to understand as human beings and as engineers that are trying to build intelligent machines, we have to figure out what tasks require intelligence and what is meant by "intelligence" in that case. Something like every day conversational, or walking, there's a whole category of things that as human beings we sometimes don't realize how difficult they are when actually have to sit down and implement that task.

Walking and every day conversation is an example. If I told you to create a chat bot, maybe as a software engineer you would sit down, "Well, that's pretty easy." Maybe you would start — I'm not a chat bot person, but it's a good example with a Turing test and so on. It's a good example of something that requires intelligence that we take for granted. If you wanted to implement to chat bot, you might say you build a giant database of different kinds of responses based on the keywords that the other person says, or you can start to try to understand the syntactical and semantic structure or the sentences.

As you start to encode all of that information you realize that the database grows the, knowledge base has to grow exponentially and you have no way of growing that automatically. As an example something when you start to actually build an engineering system, engineer a system that does this task, you'll realize how incredibly difficult that is. That's what we think of everyday conversations as an example of that.

Driving is an open question, whether driving is in that category of things that are actually exceptionally difficult and require human level intelligence or it's something that could be reduced to a problem of detecting obstacles, detecting lanes, staying in the lane and not hitting obstacles. It's an open question that we have to face. We're actually building autonomous vehicles that are supposed to drive the several trillion miles of driven every year on roads and there is only — It's tragic, but the number of fatalities in the United States is 38,000 last year. That means about one fatality per hundred or something, million miles.

That's the kind of error rate you have to achieve with these systems. You have to ask, to achieve that error rate, to achieve that kind of low margin of error, what problem needs to be solved? Is it as simple as just detecting obstacles and avoiding them or is it something where we have to create a system that has an understanding of the physics of the world, an understanding of the human intent when you're looking at pedestrians and so on. There're just these millions of factors that you have to consider. Can they be reduced to something simple?

[0:09:09.3] JM: These components of autonomous driving that you outlined in the class are perception, localization, mapping, control, planning, driver state. When you're breaking self-driving into these different components, is that taking an opinion that self-driving is false in one of the camps of being closer to chess or closer to everyday conversation or is it sort of a hedge between those?

[0:09:39.6] LF: Yeah. It's a great question. I think it's — Breaking it down into those modules means you're leaning towards, saying, it's closer to chess. That's a step towards — by chess, I mean something that's well-defined. You could formalize it. There're concepts of pieces. There're concepts of moves, actions. There are concepts of what the world encompasses, and breaking autonomous vehicles into these modules or perception, control, really, it's those two perception control is going towards the camp of chess.

[0:10:19.2] JM: Which is a problem that we're much more comfortable approaching as computer scientists at least with our contemporary techniques.

[0:10:27.8] LF: Yes. It's really well put, the contemporary techniques. If you think of software engineering, it's exactly the task of formalizing the problem, formalizing the solution, and you have to really draw out the entire thing on paper. There's no ability for systems to say, "Okay. 90% of the task would only not solve, but the system will figure it out as it goes along." You can't write a program like that in the current approaches, a program that goes out there and has a human life at stake.

[0:11:08.6] JM: Okay. I want to talk about some of the technical aspects here. Deep learning is useful for a variety of problems in self-driving cars. What are some of those problems where you can apply deep learning?

[0:11:24.4] LF: Deep learning is a type of machine learning given how successful deep learning has been really. It can be just as comfortably, you referred to as machine learning. It's most of the problems, the way it's been effective has been dominated by deep learning. Deep learning has become machine learning.

Machine learning is a set of techniques where you learn from data. A system is able to know nothing or very little in the beginning and given a lot of data and human annotation of that data. Human annotation meaning a human being steps in and says, "What's going on this data that's useful for solving the task?"

Given the human annotation and the data, you're able to learn how to behave in this world or to learn to predict things about the data, to extract knowledge from the data. That's machine learning. That's deep learning.

In the driving domain, a car, a passenger car is four wheels. Its task, it can accelerate, decelerate. It can — Go left, go right. More specifically, it can turn its steering when. It can steer. Really, those are the only actions. You can speed up, slow down, and you can steer. Then there's an external world going on around the car. Its task with perceiving that external world to the degree that it needs to understand enough to be able to accelerate, decelerate and steer.

You don't have to do any learning there. The perception can be using a sense of LIDAR and map the external environment and save that map to build a giant map with a three dimensional map of the external world and localize yourself based on that map by using this LIDAR sensor that tells you — Gives you a depth of information about around you. That's a non-learning approach.

Learning says, "I'll encode some information at the beginning, but I'm going to learn about the way the world changes. I'm going to learn about the way the world moves. I'm going to learn how the world looks visually in different situations. What is a lane marking? I'm going to learn what a lane marking is by looking at hundreds of thousands of lane markings. Having a human being step in and help me label the times when there's a lane marking. Label of the part of the

image that's a lane marking. Overtime, I'm able to generalize over videos I've never seen before of what a lane marking is."

A Tesla vehicle, for example, is one that used to use up until the end of last year, a mobile ad base system and now they have their own deep learning base system where it's predicting lane markings and there's a 100,000+ cars in the road today that are driving themselves autonomously. Tesla vehicles are able to drive for hundreds of miles in the highway and stay in the lane. It's doing exactly that with a single camera, detecting lane. That's done with machine learning.

[SPONSOR MESSAGE]

[0:14:43.4] JM: Bugsnag is an error monitoring tool that enables developers to identify, prioritize, and replicate bugs in a time efficient and enjoyable manner. Automatically capture errors and diagnostic data in any app. See errors group by root cause and focus on fixing errors that have the greatest user impact. Instantly track the performance of each application release with interactive dashboards. Automatically capture a stack trace for every error and get automatic breadcrumbs for crashes so that you can easily reproduce and fix any error.

To-date, Bugsnag has processed over 10 billion application crashes from thousands of top technology companies. To get started, go to bugsnag.com/sedaily and create an account. It takes three minutes to get up and running, an Airbns, Lyft, and Shopify are already using Bugsnag for their bug tracking and crash monitoring. Try all the features for free at bugsnack.com/sedaily.

Thanks to Bugsnag for being a sponsor of Software Engineering Daily. It's much appreciated.

[INTERVIEW CONTINUED]

[0:16:01.7] JM: We did a show recently about deep learning and reinforcement learning as applied to ViZDoom. You might have heard of that? Basically, you play your Doom, or you're a bot playing Doom and they use reinforcement and deep learning. It's a comparable problem to driving around because you're navigating a dungeon, you're evading foes or engaging with foes

in certain interaction patterns, and that show was an introduction for me to how deep learning and reinforcement learning are both key to this class of problems.

As you write, neural networks are great at memorization and not yet great at reasoning, but reinforcement learning is brute force propagation of outcomes to knowledge about states and actions. Explain what the shortcomings of raw deep learning are and how reinforcement learning can shore this up with some reasoning.

[0:17:12.4] LF: Sure. To be clear, you mentioned raw deep learning. I think the precise way to describe that is called supervised learning. Most of the success in deep learning has come from this category of techniques called supervised learning where every single piece of data that's fed into the learning system has been touched in some way by a human being to say, "What's going on in this particular piece of data?"

For images, with lane markings, it's marking the lanes in the image. For image net competition, that's marking cat versus dog. Marking for an image; is this a cat? Is this a dog? Is this some other kind of animal? That's supervised learning, and most of the successes have come from that. Up-to-date, but in terms of applications, it terms of value.

That's what I've referred to as memorization. You're memorizing at a very low grade level. You're memorizing what another human being is annotated for you. You're memorizing patterns. How, what reinforcement learning hopes to provide, in deep reinforcement learning more specifically, is to remove as much as possible the human being from the loop of teaching a system things.

The goal is to generalize over just a few examples provided by a human being in the form of a reward function or a form of an encoded model of a world that provides perhaps a simulation. Most of the success in reinforcement learning, like the Doom game, is simulation. It's not really another agent walking around in the real world. Simulation. The idea is the system learns by playing through the different ways that the world can unroll. It acts in different ways and see what those actions have impact in the external world and seeing how it can act in the future in such a way that maximizes reward.

The problem is, currently, all the reinforcement learning approaches that we know off, and this is true for machine learning in general, we have to play for large amount — For a very long time, we'll have to play with a very large amount of data in order to learn anything. If you think as human beings, when we were children, or young, or still, learning what a hot stove is like is very quick learning process. It doesn't require more than maybe two, three times to touch a hot stove before you realize when it's glowing red or its certain visual characteristics or tactile characteristics or different sensor inputs are associated with a lot of pain. That's learned really quickly.

Our current best reinforcement learning approaches and deep learning approaches have to touch that stove hundreds of thousands of times before they learn not to touch it. The problem there is we're very inefficient learners. Yes, in simulation, you can simulate the touching of the stove over and over to understand which aspects of it lead to punishment to negative rewards, but if we want to put systems in the real world, like in the driving case and to teach those systems to avoid crashes, for example, you can't quite — It's not feasible to put that system out there and crash hundreds of thousands of times, tens of thousands of times in order to learn how to avoid the crashes.

For now, there's this chase towards, "Well, can we simulate the real world to sufficiently high resolution that we're able to teach a system a simulation that's actually able to successfully act in the real world?" That's currently an open problem that hasn't been solved.

[0:21:22.7] JM: Right. You're articulating a specific case of explore versus exploit. We can't really — Like I talk to some people at Stripe, this payment company, and they will consciously let payments through that they know are fraudulent or they will sort of turn down the dial on their sensitivity to preventing fraud in order to better gather data about fraud. In that case, okay, we lose some money. Not a big deal. You can't do that with self-driving. You can't just let accidents occur and then learn from them. It's too sensitive.

[0:22:01.7] LF: Right. This is one of the biggest challenges for me, the reason I am fascinated by autonomous vehicles. It's one of the first and I believe it will be the first and the biggest way in which robots really enter our lives, everyday lives and when you begin to trust. You have to deal with issues of trust, or understanding, what a role of a robot is in a society from the

philosophical level, and the ethics side, to the technological, to everything, because the reality is if we're to design to autonomous vehicle that successfully operates in this world, it's going to have to crash. In the same way that the credit card fraud will occur. A car will have to be able to — We're going to have to be more forgiving of a car causing fatality. It's a very dark and difficult, at the philosophical level actually, to allow a machine decision to put — A human is in the hands of the machine and the machine acts in such a way that takes that human being's life. That's really difficult for us as a society to take, to understand, because especially in driving, it's such a personal experience. It's usually one human driver and there's a car, it's a machine, and here is the moment when you give control to that car and the car acts in such a way that it kills the person.

PR-wise, The New York Times coverage and our general set of view of this is we're not very forgiving. The fatality in a Tesla that happened during autopilot. It was a big deal.

[0:23:46.1] JM: I was just thinking about that. I thought the response was pretty forgiving thought. You have a car that had the autopilot on, the driver falls asleep, slams into a truck trailer decapitating him. I don't know. I felt like the public didn't really have an outcry. You didn't see people with posters in front of Tesla.

[0:24:08.1] LF: Yes, absolutely. I agree with you. The general feel I get from the public, and I've talked to a lot of people, especially in the media, and there's a hunger to attack the robots. In general, we see that robots as terminator. At any moment, they'll take over. There's a desire to find the weak link in the story, and a single fatality — I guess the point I'm trying to make is in the situation of the Tesla fatality, the car didn't actually do anything outside that looks bad in terms of the car didn't make a decision that's very poor. That, actually, it makes perfect sense even for a human being not to see the white truck. It makes sense.

What I'm more referring to is if we start allowing machine learning to be part of controlling a vehicle, there's going to be times when a lane marking is completely not seen by a car and it runs off the road. It runs off the road in such a way where any human being would have easily kept the car in control. It would lead to a fatality that looks terrible.

For example, sort of the publicize ethical issues, a car might accelerate into a crowd of people for whatever reason, whatever malfunction, perhaps in the sensor side or whatever the prediction is that the intent that it uses in detecting pedestrians, in predicting where the pedestrians go. It might accelerate into that crowd. That's a tragedy, but if overall it leads to a much safer, much slower. If it significantly decreases at 38,000 fatalities that we have in the roads today, perhaps it's something that we should be more open to. That's kind of a society we have to struggle with these questions.

[0:26:13.9] JM: I think a good preface to how weird the cases are going to be is — You have some slides in your course notes where you have these cases where you take an image of, let's say, the Empire State Building, and then you add these minor — You show it to a computer and the computer is like, "Okay, Empire State Building," and you add these minor perturbations to the picture and if a human looks at it, they're still, "Okay. It looks like the Empire State Building. The picture looks a little fuzzy now." Then you show it to a computer and it's like, "That's an ostrich." That just shows how different our perception of the world is compared to a machine, and that's why you're going to get these crazy edge cases that seem like, "How on earth did the computer not recognize the lanes there?" but if you look into the system you'll be like, "Oh, well. The model was trained in this way and it just didn't learn this edge case."

[0:27:13.4] LF: That's right. That's absolutely right. I guess I'm trying to imagine cases, but just imagine tragic cases which are very possible where a machine runs over somebody's daughter. A young girl is crossing the street and a car accelerates. As a society, we have a really hard time dealing with that concept, because, after all, an artificial intelligence system is a technology and that's heavily regulated when it's on the roads. They'll come up immediately as a discussion in public. The question is, "Well, should this technology be part of cars?" That's constantly — This comes up with drones as well and in other applications of artificial intelligence. It's definitely on our minds as we're designing machine learning algorithms for the perception side. Exactly as you said, Empire State Building being mistaken for an ostrich.

[0:28:17.4] JM: I want to talk about some lower level stuff hoping we can get into eventually. Hearing you talk about the policy stuff is really interesting and just the philosophical concerns. Continuing on that high level thread, how does human in the loop fit into the role out — Do you have a picture for how the ideal roll out strategy is going to work? I hear your concerns and

they're valid and they're very kind of hard to deal with. How do you do that? Is human in the loop the gradient that gets you off of the humans towards the full autonomy?

[0:29:01.7] LF: Yeah. I'm a big believer that the human has to remain in the loop as a supervisor and as a teacher of a car. There's mostly folks building autonomous vehicles, most of my colleagues have come from the success — Are robotics folks. Have come out of the success of the DARPA challenge, which was a challenge where vehicles were tasked with, first, going through the desert and going through an urban environment fully autonomously. No human in the picture. The car is tasked with dealing with every possible situation and not hitting things and staying in lane and all those kinds of things and not breaking down.

That's the robotics approach. I see that that, a fully autonomous vehicle, that doesn't have a steering wheel and based on the learning system or other optimization base system, is able to fully control itself. Autonomously, I agree with Warren Buffett. I think he said in 2030, 10% of cars will be fully autonomous in this way. He said that's a hedging. That's not a conservative bet. That's a risky bet. I think we're very far away from vehicles being fully autonomous.

Before that, I think human will be in the loop the way Tesla, in the case of autopilot, is now. Whenever the car can't do a situation, it has to give control to the human and say, "Holy crap! Help me out here." The big challenge here, this is the semi-autonomous vehicle approach. The challenge is the moment of, "Holy crap! Help me out here," shouldn't be one or two seconds. The transfer of control currently, like in a Tesla, is sometimes less than a second, or a couple of seconds. It depends. The red hands of shame come up as they're called. In a Tesla, where these red hands glow and you're told to take control of the vehicle, that's very few seconds. The ultimate successful system to me, you said how will this be realized, I think it would be a system that's able to 10 seconds out tell a human being to wake up, to bring their attention back to the road.

One of the missing pieces there just to quickly say is our current cars on the road today, for the most part, have no driver facing cameras or any sensors that are able to detect the driver, except the steering wheel pressure to determine that the hands are on the wheel. Your vehicle, our cars, have no idea about our state at all. Whether we're drunk, sober, sleepy, distracted, in a seat at all. There has no information about us, us personally, each individual human being, and

just general global state of where our eyes are looking, where our head is directed. That information which have a big proponent of is crucial to make a car that's able to handover control safely and give you plenty of time.

[SPONSOR MESSAGE]

[0:32:29.4] JM: Hosting this podcast is my full-time job, but I love to build software. I'm constantly writing down ideas for products; the user experience designs, the software architecture, and even the pricing models. Of course, someone needs to write the actual code for these products that I think about. For building and scaling my software products I use Toptal.

Toptal is the best place to find reasonably priced, extremely talented software engineers to build your projects from scratch or to skill your workforce. Get started today and receive a free pair of Apple AirPods after your first 20 hours of work by signing up at toptal.com/sedaily. There is none of the frustration of interviewing freelancers who aren't suitable for the job. 90% of Toptal clients hire the first expert that they're introduced to.

The average time to getting matched with the top developer is less than 24 hours, so you can get your project started quickly. Go to toptal.com/sedaily to start working with an engineer who can build your project or who can add to the workforce of the company that you work at. I've used Toptal to build three separate engineering projects and I genuinely love the product.

Also, as a special offer to Software Engineering Daily listeners, Toptal will be sending out a pair of Apple AirPods to everyone who signs up through our link and engages in a minimum of 20 work hours. To check it out and start putting your ideas into action, go to toptal.com/sedaily.

If you're an engineer looking for freelance work, I also recommend Toptal. All of the developers I've worked with have been very happy with the platform. Thanks to Toptal for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[0:34:33.9] JM: Engineers who are listening to this podcast, often times, they tune in to get a high level view of how to build certain things. If you're building a web app, you would learn model view controller, or client server. With a self-driving car, I don't really understand how the different systems fit together. We talked about these components; perception, localization, mapping and so on. Is there sort of best practices for how these system are put together today? Can you, I guess, give me an overview for the software architecture of a self-driving car?

[0:35:17.5] LF: Sure. There's no best practices, first of all. It's such an open world. There are so many different approaches in terms of from the very subtle, the thousands of parameters that control every algorithm, to just how the pieces fit together, which sensors are used, and so on? There's just a lot of variety.

In terms of the abstract classes of the different approaches, is first, you have to have an ability what's called SLAM, simultaneous localization and mapping. You have to be able to find out where you are in the world. As a car, you need to be able to position yourself in the three dimensional world. That can be done in a bunch of ways. One is having already — First of all, it can be done with having a pretty good map of the world, like Google Maps, and having a GPS. That's one very crude estimate of where you're located, GPS location, latitude, longitude. It will tell you where you are in that map. You can do some basic information. Which road I'm on? How close am I to the intersection? Which lane I'm on? Obviously, GPS is very noisy. That's one estimate.

There could be another estimate. You build that separate system. As a software engineer, you would build that system that says, "I'm going to." The input to the function is my GPS coordinates and I have a database, that's a Google Maps, and it puts me to those GPS coordinates and says, "Okay, I'm in the left lane in this road and I am 100 meters away from this intersection." That's one little module.

Another module will be, "Okay, I'll have a stereo vision camera," which means there's two cameras positioned close to each other looking out into the external world and that camera is able to locate the different obstacles, objects in the environment and estimate crudely their depth, meaning the three dimensional position, the point cloud of where they're located; the pedestrian, traffic lights, signs, and so on. That's another little function that says that input is two

images coming at 30+ frames a second. The output says, “I am located at this position relative to the traffic sign, the pedestrian, other objects in the scene. Triangulating everything together.

If I have implemented both these functions, I have two pieces of information that says my GPS coordinates and my approximate triangulation of where I’m located relative to those objects. That can fuse those together, decision fusion, or sensor fusion, data fusion, where you take those two pieces of information and combine them to make a slightly better estimate about where you’re located in that world. The whole point is to be able to locate yourself as accurately as possible in the semantic structure of what this task is. Driving task, you have to be in be lane. You have not be hitting things, other obstacles in the scene, and you have to be navigating, moving around to get to some location.

In that sense, part one is you localize yourself. It’s plug and play. Another developer might come up and say, “I have a LIDAR, and that LIDAR will be able to do a much better job than your stereo vision camera to give you a better position in this world.” We can all just combine those functions together with sensor fusion and chase the more and more and more accurate localization.

The second part of driving is once you know where you are, you want to — And where the other objects are. You want to move around those objects. That’s where the control planning happens. The way you do it is you have a three dimensional model of the world with some degree of uncertainty and you generate thousands of different trajectories for yourself and the trajectories for others that you believe based in your model how other objects move, other vehicles, other pedestrians.

Based on all those trajectories, it’s an optimization problem for the optimization based approach to pick which is the best trajectory. For the learning best approach, you use something like reinforcement learning to say, “What is the best trajectory around those objects?” That’s it. That’s driving. Perceive the world, and then move in that world.

[0:40:21.2] JM: When you’re teaching the course in self-driving deep learning, what have been the concepts that are most difficult for the students to grasp?

[0:40:33.2] LF: That's a good question.

[0:40:36.0] JM: That's a tough question. Talk about how you sequence the course in self-driving deep learning, or talk about how if somebody's listening to this and they want to know how to get started. Obviously, there're courses. I think there's Coursera course, but you maybe you have some abstract suggestions for how to strategize about the learning process.

[0:40:59.9] LF: Right. I think it's just a really good question to say what's the struggle the most, and just to answer it quickly, is I think as not just students, but Ph.D. students and research scientists, professors, everybody, is struggling with the question of why something works. On the very basic, feet forward, fully connecting you on that works to all the different flavors in neural networks, convolution neural networks, or current neural networks for deep reinforcement learning, figuring out why something works so that when something doesn't work you know how to fix it.

Why there's a bunch of parameters that control the behavior of a neural network. What do I set each individual parameter to make it work for my particular application? Gaining an intuition, this is the process of learning, I think, in the case of deep learning course like the one I taught, like the other out there, is gaining an institution about what works and what doesn't. How many nodes in a network? How many layers? What kind of optimization algorithms? What kind of learning parameters? Learning rate? What kind of preprocessing on the data is needed? What kind of — There's a lot of different parameters that control, especially reinforcement learning that control the training process.

The way you teach and the way I learn myself is really boiled down every single type of approach, the simplest possible problem. Look at a very small neural network and learn how — For example, I was looking at back propagation, which is the process of training a network, a network that is producing random results. How do you take that network and, overtime, train it to produce the correct results? That's the process of back propagating, back propagating errors through the networks such they just waits not to make those errors.

You can start with a network of 100 million parameters, or you can start with a network with just 10 parameters. To visualize and play around and see how different hyper parameters of training

a network affect the way the waits are adjusted. Look at toy problems for each one to slowly gain the intuition, because the problem is when you scale the size of a network, that's where the "magic happens", but we don't have a good intuition about why it happens. Why in the case of deep reinforcement learning, a system that knows nothing is able to learn overtime how to play an Atari game is a mystery. It would see through this kind of brute force process of reward and punishment. The ability to learn abstract notions of what it takes to win a game would seem to be impossible, but it works. Overtime, you start to gain an intuition of why it might work. That's the challenge.

[0:44:18.6] JM: What's funny about this is I used to be a poker player back in the day, back when it was easier to win money playing poker online. One of the things — The people who are successful were the people who read these books about mathematics, poker math, or they could articulate — Many of them could articulate mathematically why a play was good. Why a strategy was useful. We always used to mock the people who were instinctive players and like, "Oh, I've got a feel for this thing."

Some of the players who would play by feel, they would be successful. Overtime, the people who are mathematically oriented just dominated them. They always had a reason for why they did something. It's just funny that, now, you're talking a branch of computer science where, basically, we don't really know how to talk about this formally. I've heard multiple say this, like, "You have to develop this intuitive feel for what's going to work in deep learning." Why is that? How did we get here?

[0:45:34.9] LF: It's interesting. Because, yeah, in the case of poker, the math guys and girls — Probably, they succeeded because they're able to silence the irrational parts of our brain that says, "I have a feeling that the flush is coming or something." You're able to look at the actual odds and calculate statistically what is the right move here, and based on that information, start to play with the bluffing and all that kind of stuff.

It's hard, because that's the scientific way. To give that up, to explore how a system, how intelligence can emerge requires some intuition. You have to let go of the need to prove stuff. I come from the theoretical computer science background where you have to prove P versus NP. You have to prove the running time complexity of an algorithm. The idea that you have no way

to prove, but it just works, was viewed very negatively in academia when I started. I think that's changed completely with the success of deep learning.

It's actually the reason why deep learning — Deep learning is just another word for neural networks. The reason why neural networks have gone through a couple of winters, AI winters, that because we don't understand why they work but they seem to work. It captivates the minds of people and they get super excited. They start thinking that they will solve everything in the world. We have created general intelligence. That was true when the first Perceptron came out in the '50s, I think. They're going to solve everything. They're going to fly to the moon. Flying cars, for sure, will be around.

The reality is, "No. These are just effective tools," and we're slowly, slowly trying to discover in the same way that Darwin sailed around the world discovering what the hell is the mechanism behind the evolution, the diversity that we see in the world. That's how we're trying to sail around the neural network world and trying to understand what is the mechanism behind the emergent ability of these networks to predict stuff, cat versus dog at the simplest level. The coolest and most difficult level is the reinforcement or even unsupervised learning, how you're able to find generalized patterns from nothing, essentially, from no dictionary, no supervised learning human input.

[0:48:23.4] JM: Okay. I can talk to you for a long time about this, but we're running up against time. We're drawing to the close of the interview, and I wanted to ask you some about the business of self-driving and how these companies are strategizing about it. How they're thinking about it. I read that you're interested in jiu-jitsu and you practiced jiu-jitsu and I think of the strategy of Google as very Jiu-Jitsu in the self-driving world because it seems like they're very much trying to lay as low as possible. I don't know. How do you see the different players in the self-driving space and how do you contrast their strategies? Do you have any predictions about how they're going to roll out things?

[0:49:13.9] LF: It's interesting. I used to work at Google, and I now work a lot with Teslas, or Teslas, and it's a very different approach of Elon Musk and Google are way mono. One is, yes, Google is way-mo. The approach is to stay low, to stay quiet. Design vehicles that are very safe,

never hit anything. They map at a very high resolution, the world around them, and travel at 35 miles an hour, at 40 miles an hour around a very well-mapped safe street.

The Elon Musk approach is to say, “Basically, everybody in the auto industry is scared, and I’m going to — Just like you go to Mars, I’m going to release fully autonomous vehicles by next year and say, “We’re going to do it. We’re going to shoot to the moon and get it done.” That’s his approach and how shy away and, on Twitter, promise things. Literally, like I said, there are 100,000+ vehicles are driving themselves autonomously on the roads today. This is sort of the little known secret of Tesla is autopilot, for those that haven’t driven it, is really pretty advanced in terms of automation capability. Those are the two approaches.

I am generally a fan — In terms of just personally speaking, I’m a fan of the bold and the brave and the risk takers in the world. I believe that in order to get reduce from 30,000 fatalities to zero fatalities, we have to take risks. We have to be brave. We have to be willing to lose our entire company and idea, and that’s exactly what we have to do with autonomous vehicles, because if your car kills several hundred people from a malfunction, there’s a very big risk you’re taking than PR or government stepping in and shutting down the company and so on. I’m a big fan of Tesla in that way, but I’m also because we’re working closely with NHTSA, which is a government organization tasked with ensuring that vehicles on the road are safe.

I also am well-aware of safety and how important that is. There’s a lot of my colleagues in computer science, sometimes are so enamored by the coolness of a technology that they don’t think about the safety implications. This is sort of the worry that people have about creating artificial intelligence that us, nerds, will create this awesome AI and forget to make it ethically sound, to make sure it makes safe decisions, ethically sound decisions. All that said, I really despise caution and being careful, is I think the whole idea of science and innovation is being brave, bold in these ideas.

That said, Google, I think — That’s one side. It’s just me personally, sort of on a horse with a sword. In terms of business side, what is the right bet here? I think Google has been exceptionally successful at playing the long game, and I think Google has a lot more to lose than Tesla. I think, as I said, autonomous vehicles will not be here until — I’m with Warren Buffett. He’s rarely wrong. I think they won’t be here for a couple of decades. As a business

decision, it's not a good one to go all in on autonomous vehicles at this time. Even though the public is generally very hopeful and optimistic about it — There's been a lot of companies that promise in 2020, in 2022, in 2025, we'll have fully autonomous vehicles, Toyota, Ford, everybody, and their grandmother is promised that they'll have fully autonomous vehicles.

My bet is that the realities will be a farther out because of all the different, the technology side, the policy side, the societal side. Just all of those are huge challenges to overcome. To overcome them, I believe we need the risk takers. I'm a huge supporter of our friend, Elon.

[0:53:45.5] JM: Yes. Okay. Last question to wrap up, deep learning and machine learning, these fields are so rich. There's so much going on in them. There's this chart you have in one of your presentations that I really like that shows, basically, that the trajectory of a person who is learning about deep learning. You start off like, "Oh, this is so intimidating," then very quickly, you're like, "Oh, I know everything and this is going to be awesome," and then it drops off again and you realize you know nothing about the field because it's so deep and there's so much going on. Do you feel like we're witnessing a field that's going to bifurcate away from computer science because it is so deep?

[0:54:46.5] LF: That's a really interesting idea. That's really, really interesting. I haven't actually heard that articulated in that way before. That's a feeling I actually have. The experience I have is there's a lot of folks, there's so much interesting. There are young students coming to me every day saying, "I'm fascinated by learning." They're not talking about I'm fascinated by databases or JavaScript frameworks, or the software engineering concept, or theoretical computer science, big-O, or system architecture.

It almost feels like there's this very hot topic that's actually counter to the way computer science is taught. It's a fascinating idea that you would have a deep learning engineer that's completely —

[0:55:39.3] JM: Because you don't need to know databases. You don't need to know model view control. You don't need to know big-O. You don't really need any of these stuff.

[0:55:45.5] LF: Of course.

[0:55:48.5] JM: It helps. It doesn't hurt.

[0:55:51.4] LF: You need to know all that stuff if you want to innovate the future of deep learning architectures, but you're right, this sort of — There's a lot of companies out there that sort of need to solve practical problems with machine learning. For that, that's essentially — That's kind of called data science now. it's when you basically are cleaning datasets, organizing datasets and extracting information from them using various methods, statistical or machine learning. Really, that could be some discipline. That's a really fascinating thing to think about, but I hope not, because I hope computers — It's a weird thing to think about because computer science is infiltrating everything. It's mechanical engineering, electrical engineering, aerospace. Every single field — Library sciences, psychology.

Psychologists now all have to code. It's the people in psychology who aren't able to program are finding themselves left behind because most — With the advent of mechanical Turk, you want to be able to use computers to put together cool experiments. It's hard to know what computer science will be in 30 years, because software engineering is becoming the core of every discipline. That's a really interesting question to think about what that evolves, where robotics fits into this whole picture too. Yeah, it's a great question. I think we're going to be doing deep learning — If you're a literature major college, I think, in 20 years, you'll probably be using TensorFlow for your homework.

[0:57:41.1] JM: All right. Lex, it's been really fun talking to you. The time flew by.

[0:57:44.3] LF: Yeah. Thanks, Jeff. Thanks so much.

[0:57:46.0] JM: Okay. Cool.

[END OF INTERVIEW]

[0:57:49.9] JM: VividCortex is the best way to improve your database performance, efficiency, and uptime. It's a cloud-hosted monitoring platform that eliminates your most critical visibility gap, providing insights at 1-second granularity into production database workload and query

performance. It measures the execution and resource consumption of every statement and transaction, so you can proactively fix future database issues before they impact customers.

To learn more, visitvividcortex.com/sedaily and find out why companies like Github, DigitalOcean, and Yelp all use VividCortex to see deeper into their database performance. Learn more atvividcortex.com/sedaily, and get started today with VividCortex.
[END]