EPISODE 96

[INTRODUCTION]

[0:00:10.8] SC: Hello and welcome to another episode of TWiML Talk, the podcast where I interview interesting people doing interesting things in machine learning and artificial intelligence. I'm your host, Sam Charrington.

Last week, I spent some time at CES in Las Vegas exploring the vast sea of drones, cameras, paper-thin TVs, robots, laundry-folding closets and other smart devices. You name it, it was there. Of course, I was also able to sit down with some really interesting people working on some pretty cool AI-enabled products.

Head on over to our YouTube channel to check out some behind-the-scenes footage from my interviews and other quick takes from the show. Of course, be on the lookout for our AI and consumer electronics series right here on the podcast coming soon.

A quick note, this is your final reminder about tomorrow's TWiML online meetup. At 3 PM Pacific, we'll be joined by Microsoft researcher Timnit Gebru. Timnit will be joining us to discuss her paper, Using Deep Learning and Google Street View to Estimate the Demographic Makeup of Neighborhoods Across the United States. I am specially looking forward to her going into more detail about the pipelined she used to identify 22 million cars and 15 million street view images.

As usual, we'll get the meetup kicked off with a discussion segment in which we'll be exploring your AI resolutions and predictions for 2018. For links to the paper, to register for the meetup or to check out previous meetups, visit twimlai.com/meetup.

The show you're about to hear is part of a series of shows recorded at the RE•WORK Deep Learning Summit in Montreal back in October. This was a great event. In fact, their next event the Deep Learning Summit San Francisco is right around the corner on January 25th and 26th. The summit will feature more leading researchers and technologist, like the ones you'll hear here on the show this week, including Ian Goodfello of Google Brain and Daphne Koller of Calico Labs and more. Definitely check it out and use the code TWIMLAI for 20% off of registration.

In this episode, we hear from David Duvenaud, Assistant Professor in the Computer Science and Statistics Department at the University of Toronto. David joined me after his talk at the Deep Learning Summit on Composing Graphical Models with Neural Networks for Structured Representations and Fast Inference.

In our conversation, we discussed the generalized modeling at Inference Framework that David and his team have created, which combines the strengths of both probabilistic graphical models and deep learning methods.

He gives us a walkthrough of his use case, which is to automatically segment and categorize mouse behavior from raw video. We discussed how the framework is applied to this and other use cases. We also discuss some of the differences between the frequentist and Bayesian statistical approaches.

I had a great time with this interview and I think you will too.

Now, on to the show.

[INTERVIEW]

[0:03:34.6] SC: All right, everyone. I am here at the RE•WORK Deep Learning Conference in Montreal. I am with David Duvenaud. David is a professor at the University of Toronto and he actually just got off stage. I'm excited to be sitting here with him, or here with you I should say, David. Welcome to the podcast.

[0:03:52.8] DD: Thank you, Sam.

[0:03:54.0] SC: Tell me a little bit about your background and how you got interested in machine learning and AI. It sounds like you come a little bit from the statistics side of the world?

[0:04:02.0] DD: No. I've actually been maybe pushed into the stats. I mean, I'm have appointed stats and CS at the University of Toronto and it's really nice. Now, actually the way I got into this was actually reading [inaudible 0:04:14.6] webpage when I was an undergrad. To his credit, he really was thinking about a lot of things that people later also found interesting. He has a way of presenting his ideas as the last word. I mean, everyone does.

At the time, I didn't really know how to evaluate these things and I thought, "Oh, man. I need to go to grad school because this going to solve AI in like two years or so." Yeah. I mean, of course now I like, I have much credit view and I –

[0:04:42.0] SC: Did you go to grad school in Switzerland?

[0:04:43.8] DD: No. No. Although I certainly like [inaudible 0:04:45.5]. I actually went to grad school at The University of British Columbia. Initially, actually all my friends went to grad school and my first time I applied for scholarship I got rejected. I had a big shift on my shoulder and I was like, "Grad school is for jerks."

Yeah, then eventually I applied again and got it. Yeah, went to work with Kevin Murphy back when graphical models were still cool and we did a bit of machine vision and things like that. Yeah, and then he actually set me up with an internship at Google. It wasn't called Google Brain back then. It was actually the video content analysis team, but it was all the same people that ended up forming it. We just tried to get covenants to predict whether YouTube videos contained people dancing for these contents that YouTube is running at the time.

[0:05:29.3] SC: Interesting. Interesting. You're now at University of Toronto?

[0:05:33.0] DD: Yes.

[0:05:33.9] SC: How long have you been at the university?

[0:05:35.4] DD: Just a year actually.

[0:05:37.8] SC: You said that you've been pushed into this joint appointment?

[0:05:42.1] DD: Well, okay. Let's answer to that. I just mean that I started off being interested in machine learning and then went and did – my PhD on Bayesian and parametrics. I realized that math was the missing component, like everyone in machine learning has to build a code and do math, but you really can't get away without knowing the math.

It really, a little bit for most and can go a long way. Especially recently, I've been working on building gradient estimators for discrete latent variable models. This also shows up in

reinforcement learning, where having unbiased gradient estimators is actually really important. Unbiased sounds like this boring horrible thing that frequency statisticians care about, and it is. But it actually now I think is going to turn out to be a central thing that people are going to be worrying about over the next few years.

[0:06:32.3] SC: You just did your talk and your talk was on combining graphical models with deep learning models?

[0:06:39.8] DD: Yeah.

[0:06:40.8] SC: Tell us a little bit about the talk as an overview and we can dive into the different elements of it.

[0:06:46.6] DD: Sure. Well, the way that paper that came about was actually pretty fun. I just did a post doc at Harvard with this guy Matt Johnson, who is now at Google Brain. When I showed up, I had just de-converted from Bayesian non-parametrics where I was trying to fit giant infinite dimensional graphical models to everything.

I had realized the limitations of these approaches and I was really excited about variational auto-encoders where we learned a generative model using a neural network and we also used a neural network to learn to do inference in that model. As I said in the talk, people have been able to write down really rich generative models for a long time, but they haven't been able to do inference on them, and that's the limiting factor.

With regards to this paper, Matt was coming from also old-school background where he was saying, "Oh, I want to use graphical models, linear dynamic systems," like nice models that you can analyze and understand. I was telling him, "No, no, no. You got to forget all of that. You have to just use neural networks to do inference, so we can fit everything with a variational auto-encoder. We had this long back and forth over the course of about a year. In the end, it was like my chocolate in your peanut butter and we had this nice synthesis of ideas. We said, "Maybe we can combine these and get the best of both worlds."

[0:07:57.9] SC: Okay. When you say graphical model, what do we mean with that? What do you mean by that?

[0:08:03.3] DD: That's a pretty broad phrase. It basically just means –

[0:08:06.4] SC: It doesn't mean graphics as we traditionally talk about graphics. That's the starting place, right?

[0:08:10.4] DD: The graphical and graph the models means like graphs. It is that initially when people started writing down models, they just wrote down equations that would say what the probability of different things were. It was pretty hard to analyze these models.

I think, like Kevin Murphy, Daphne Koller, Nir Friedman. These guys said, "Ah, what if we actually represent the dependence of different variables as a graph?" The idea is that if I say that smoking causes cancer, then I would have a arrow going from smoking to cancer.

This says something about how the probability of these things change together. If I get cancer, that doesn't make me smoke, but if I smoke then I may get cancer. Once we start to have 10 or a 100 different variables keeping track of all these arrows or these relationships becomes pretty tricky. But when we expressed these models as a graph, we can automatically analyze them and ask whether we think that drinking wine causes better test scores through some complicated mechanism, or we can also ask what information would we need to learn in order to tell us what the answer is.

Yeah. This is this – it's not old-fashion AI, but it's this we're going to have this rational understandable models where every human concept has a little box that we put it in and we try to understand everything that's happening in our models.

[0:09:30.9] SC: Okay. When we talk about graphs and applying graphs to these types of models, so one of the things that I think of at least is some of the deep learning frameworks, like Tensorflow allow you to create a neural network architectures using graphs, but that's like at a higher level than what you're talking about, right?

[0:09:48.6] DD: Yeah. That's a confusingly similar idea, because we write down this computation graphs where we have these arrows that mean A is a function of B. That's also the same relationship we're talking about when we write down graphical models. In graphical models, the relationships are all probabilistic, like we're not saying that A determines B. We're saying the probability of B depends on A.

Then the thing we can do with graphical models that we can't do with neural networks so easily is to go backwards and say, "Given that someone has cancer, what is the chance that they smoked?" Which is something – well, it's not as easy to run a neural network backwards. When we run a graphical model backwards, we're asking what are the hidden causes of the things that we observed?

This is what we tend to refer to as the inference problem. How do we figure what – how does what we saw change what we believe about what we didn't see? As an example, you can even view for instance like learning grammar, like when babies learn language, they hear all these sentences and there's this hidden thing, which is grammar and vocabulary and all these rules about how language goes together.

The problem of hearing a bunch of sentences and then trying to figure out which are the likely rules of that language is an inference problem. This is why people have been really excited about these generative models are like also called laying variable models for a long time. Yeah, like it is the grammar, it's this latent unseen variable that we have to infer.

Also, this motivates why people have been so excited by inference. If you go to NIPS, there is the advances in approximate vision inference workshop and variance of it. It's one of my favorite parts of NIPS, because it sounds really boring and dry, but this – the inference problem is the bottleneck for doing all the cool things that babies can do that we can't do with at least – that's at least one bottleneck. Even if we told inference, there's probably more problems that we need to solve.

For instance, people like Josh Tenenbaum at MIT for now like 20 years or so have been –

[0:11:54.0] SC: I need to get him on the show. His name has come up probably like three times, just today.

[0:11:58.7] DD: Yeah. He's just a really inspiring person to talk to, because he really saw this vision a long time ago that, "Guys, guys, guys. If we could just figure out how to do inference, we would be able to not only explain how humans do all these things, but also get machines to do them ourselves."

He's been looking into these for a long time. I think actually all the – I would be sure to keep an eye on the stuff that's coming out of his lab, because inference methods have just been making these major leaps in balance in the last three or four years.

[0:12:30.4] DD: I think inference methods is maybe another name collision, because we refer to inference as using models generally. But again, we're talking about something different here. We're talking about –

[0:12:43.5] DD: Yeah, maybe I would say probabilistic inference.

[0:12:45.5] SC: Yeah.

[0:12:46.5] DD: I agree that this word is completely overloaded, and also in frequentist statistics that has another meaning.

[0:12:51.0] SC: In frequentist. What is frequentist statistics? I can't even say –

[0:12:55.7] DD: You can't really, because – so I would come myself with Bayesian, so it's like asking like unable to describe a Republican, or like can you describe a Protestant or something. They're interested in worst-case guarantees, computing P values, doing hypothesis testing, basically making procedures that can tell us what the data say about this particular question, regardless of what we happen to think before.

Then the Bayesian science says, "Well, let's actually ask how combine what we saw today with what we need before." I mean, actually that's the centered answer. To me, the real answer is Bayesians consider all possibilities and they just keep all them around them and weight them all equally. Or not equally, weight them according to how well they fit the data.

Figuring this, trying to identify the best possible hypothesis. These are like – there is good reasons to do both. These are very different schools of thought. There's been some fortunate bit of travel and I think where people naturally tend to form groups. Yeah, so that has definitely happened and that's been a thing for 70 years and so it's the sixth now.

[0:14:06.1] SC: Yeah, maybe a digression are there a set of things that you think of about statistics that you wish more people doing machine learning or deep learning, knew or understood better?

[0:14:18.2] DD: Okay. Well –

[0:14:19.9] SC: I'm going to take that as a yes.

[0:14:22.3] DD: There's like a little rent that I've been wanting to go on for a long time, which even like princess at Harvard when they were teaching machine learning someone said, "What's the difference between a frequentist and of Bayesian?" Then they said, "Oh, frequentist treats the data as around a variable and the Bayesian treats the truth as around a variable." That is technically what is happening, but it's just like a bizarre miss-framing of the entire discussion.

I mean, like I ran a variable. It's not necessarily random and it's not necessarily a variable. It's a very bad name, but the idea is that the frequentist is going to say if this thing was the case, how likely would I have been to see the data that I saw? That's a sort of – also just said it's often done. Also it's a method.

The Bayesian will say, "Given that I did see this data, what do I believe about the truth?" Even though I think the truth is fixed and it's not random, because I don't know it, I can use probability to describe my state of uncertainty. That doesn't mean that I think it's random. That just means I am using probability to describe my uncertainty. Yeah, so that's my public service announcement.

[0:15:27.7] SC: Okay. That's describing these two tribes. Are there ways that you see those two schools of thought influencing the way folks approach machine learning and AI, that particularly that you think a little bit more commonality or something with advances as a community?

[0:15:53.1] DD: Well, it's funny because deep learning has represented a de facto third direction, or maybe a synthesis and these debates have been Bayesian and frequentist methods, I think really took a backseats to saying, "Let's just define a probabilistic model that would give us a continuous loss function that we can optimize and use gradient-based optimization to do maximum width estimation."

[0:16:17.9] SC: Deep learning in a nutshell.

[0:16:19.2] DD: Deep learning in a nutshell. This is takes elements for both. Now of course that people are saying how do we train deep learning models with less data. People are looking more into Bayesian deep learning, which actually can be made to look a lot like standard deep learning where you just a little bit noise to everything and just really nice.

Yeah, I guess the thing is that most of the classic frequentist methods are based on taking really simple methods that are easy to analyze and prove things about, just sort of developing those. Whereas, for neural networks you can't really say much about them in the hard on proving asymptotic sense that you could with maximum like width estimation, or okay it sounds –let's say with like, these simple estimators people like to use in science.

People used to love these frequentist estimators, because they were simple and fast. Now we accepted that if you pay this price of having a little bit more complex models and maybe a GPU or something. You're going to get better enough performance that you'll just forget about any asymptotic guarantees that the other models might've had.

[0:17:22.4] SC: What are some examples of frequentist models?

[0:17:25.7] DD: Frequentist methods, let's see. There was this whole [inaudible 0:17:29.1] industry of frequentist kernel methods that would say, let's define a non-parametric estimator by considering all possible functions in some infinite dimensional hubric space that could possibly separate our data, or explain it, or model its density.

These methods still have a place and I think one of the – like Earl's presentation at NIPS the series is on its methods. They're not done, but they're definitely in a little bit of – there's definitely like a kernel winter happening right now.

[0:18:02.1] SC: Okay. Then the – in fact, your presentation was talking about in some sense how to combine elements of both of these schools of thought?

[0:18:13.7] DD: Yeah, exactly. I think when I got to Harvard I was like one of these people who recently de-converts from a religion and they have nothing that – any bad to say about it. That was saying, "Well, are you sure you want to really throw this all out?" We really did have this

problem of analyzing this most data and it had fit just like the pure graphical model standard approach to the theta and it had a bunch of problems with –

[0:18:39.1] SC: Well, let's hit pause there and talk about the data so that we're all on the same page.

[0:18:43.6] DD: Sure. The data is a bunch of connect video of mice running around the dark. The idea is that when biologist want to measure what happens when we change the genes of a mouse, or give it a drug, or expose it to the odor of a fox or whatever, they need to quantify how its behavior has change, so they can write about how our model of autistic mice do these more then. But then when we give them this drug, they act more normally or something like that.

[0:19:09.6] SC: I couldn't tell from the video whether that was top-down, or through a glass for looking out or something like that?

[0:19:16.9] DD: It's top-down.

[0:19:17.5] SC: Top-down? Okay.

[0:19:18.9] DD: The idea is that right now they have a army of grad students who spend thousands of hours watching this video and then saying, "Okay, now the mouse ate something and now he ran over there and now he stood up and now he went over to his buddy." This is cruel both to the students and it also introduces variability between different people who might –

[0:19:41.2] SC: Labeling error or something along those lines?

[0:19:42.7] DD: Yeah. Just differences, right? It's hard to canonically say like, okay this is a mouse that is grooming or no, right? Also, even –

[0:19:50.1] SC: Labeling noise more so than ever maybe? That's different interpretations of what the mouse is doing at a given time.

[0:19:56.4] DD: Yeah, exactly. Really, we can imagine this changing systematically across labs, maybe in different countries. Maybe it's hard. A language barrier maybe it could mean something solely different. We'd really like to automate this part of the scientific pipeline, both

just because it will save us time. But also because it will help us do better science by removing one, or by standardizing one part of the entire pipeline.

[0:20:21.5] SC: You have this data – were there a fixed number of classes or activities that you were trying to capture, or was part of the challenge trying to identify how many fixed sets of activities there were?

[0:20:36.2] DD: Yeah, great question. We really wanted to make sure that we didn't have to tell it exactly how many different classes of activity that were, because that would defeat the purpose and it will also make you wonder, "Well, what if I had given it more class or one less? The results have been totally different."

This is one of the benefits of being patient is that you can compare the model fit in a systematic way between different models. What we did was we said, "Okay, there are up to –" I think we chose 40 different clusters. The idea was that we –

[0:21:11.9] SC: 40 clusters or 40 classes within –

[0:21:15.0] DD: Sorry. Well, classes and clusters are the same thing in the way that I'm talking about this.

[0:21:20.2] SC: Okay.

[0:21:20.8] DD: Then the idea is that we could let the model choose how often different activities appeared. Some of them, it would just never use and the idea was that we tell there are at least 40 clusters and it will say, I can explain this data with only 20. The other 20 just stay off forever, so automatically learning the number of clusters. We just have to make sure we have a good upper bound on how many there could possibly be.

[0:21:45.5] SC: I'm sorry, you mean clusters in a sense of cluster data points as that will define a class? Is that the right what I think about this, or?

[0:21:53.2] DD: That's a really good point. I guess, I mean clusters in a more abstract sense where we're clustering the dynamics of the most movement. The idea being that a behavior is not a particular post and that must be it. It's not a particular post that the most might be in, but

it's the way that he moves from one post to another, or when he's grooming he is like moving in this circular motion, or something like that. I mean, a cluster and the dynamics is one behavior that he could be doing.

[0:22:23.1] SC: Okay. How many of these clusters ended up being identified?

[0:22:29.0] DD: To be honest, I forget and I think it was around 20. I would have to say Matt was the first off there. He's the one who really spent a lot of time with the data.

[0:22:38.0] SC: Okay. How did the graphical analysis or the graphical element of this play in?

[0:22:46.3] DD: Yeah. The alternative, the baseline that we could've done would just be to say that there's some recurrent neural network that defines how the most changes through time. Then we'll have some continuous vector that is changing and we don't necessarily really know what that continuous factor means.

That would've probably fit the data pretty well. We would've been able to predict the – most as future movements I think about as well, but we wouldn't have any to look in and say, "Oh, there is these distinct clusters." That was the whole motivation was the interpretability of this model.

[0:23:23.1] SC: What's the process for building a graphical model? Are you literally identifying states in transition vectors and lost things like that, or is it more abstract than mathematical?

[0:23:36.7] DD: Well, we try to make it as abstract as possible, because we want to let the data speak for itself as much as possible. All we did was we said, there is 40 different states the mouse could be in. There is some probability of transitioning from each one to each other one, and we don't know what that is, so the model has to learn that. It also has to learn how the states influence the dynamics and it also has to learn how those – like the most as body say trans – corresponds to actual video frames.

The idea is that all we basically said is there is some discrete stuff that controls some continuous stuff, that controls some video stuff and all the connections between those things and the connections to time had to be learned automatically.

[0:24:18.1] SC: Did you end up finding that – I'm thinking about the density or sparsity of the connection graph. Does that play a significant role in this? What did it end up looking like?

[0:24:30.0] DD: Great question. I do think that mouse's behavior transitions probably are sparse. Maybe he never goes from eating to standing up right away or something like that. We didn't actually put any capacity for the model to – or we didn't put any prior information about whether it's a transition matrix were sparse or not.

To be honest with you, it didn't look at how sparse the learned matrix was. I bet it was sparse. Yeah, so these are the refinements that we would deem like to make. Actually I want to say, these are the refinements that we would like our learning algorithm to be able to propose on its own. At the end of the talk I said, so right now we built the model, but what if we got it wrong? What if most behavior isn't discrete, or what if – when there's two mice involved, there's some more complicated structure that we just don't understand most theology, so we don't even know how to write it down.

What we'd really like to do is try learning both, all the parameters of these models and which types of structure they should have as well. There's no technical reason why we can't do this, it's just that it requires searching over this discrete space, which is always a big pain.

[0:25:39.2] SC: Are there things happening in the field that you think will enable you to do that? Is it just going to be brute force, better compute, or are there folks doing research that you think will lend themselves to – or what are the research areas that will lend themselves to figuring this out?

[0:25:59.0] DD: Right. I'm glad you asked. I perfectly think that in the next few years we're going to see a lot of progress and models of how to fit discrete models, or in methods of discrete, sorry. I think we're going to see a lot of progress on methods to fit discrete to models. The deep learning revolution has basically been all about continuous everything; we have continuous parameters, continuous predictions

We have a latent variable model; the latent variable are mostly continuous. The stuff that I talk about today was just a tiny step. We added easy to handle discrete-latent variable. But really the stuff that I think is more interesting is going back to the grammar example. Learning an

entire grammar on a language, or even learning a parse tree for a given sentence is this complicated discrete objects that you can't even say it's one out of a 100 possibilities. It's actually like these entire trees of rules.

There still isn't a much better way to handle these sorts of models than we had 10 years ago. One thing that I have been really excited about and getting my students to work is how do we find continuous relaxations or gradient estimators for models with discrete latent structure. This is going to let us – I mean, if it works, do all sorts of things, like learn hard attention models, train GANs to generate text.

Yeah, as I said they're in grammars learn models with these interpretable structures, learn to do things like produce programs. I mean, obviously none of these stuff is going to work out of the box. But again, gradient-based estimation is a really, really great method because it scales to millions of parameters in a way that something like evolutionary algorithms I think never will.

There might be another way forward, but the way out for it I'm excited about is trying to get good gradient estimators for ML as with discrete structure.

[0:27:51.8] SC: Okay. Just so I can make sure I understand, the single latent variable in the example that you presented is the cluster that the mouse is in at a given time?

[0:28:05.8] DD: Exactly.

[0:28:07.6] SC: It's discrete because it's a cluster you quantize it inherently?

[0:28:13.4] DD: Yeah, exactly.

[0:28:13.8] SC: Got it. Are there other formulations of that same – I mean, there are lots of formulations of that problem that aren't necessarily discrete. You went down this particular path. Why?

[0:28:25.6] DD: Again, just for interpretability. The idea is yeah, we could've said that there is 10 actions that he can be doing to some degree. Maybe he's eating a little bit, maybe he is running a little bit. Then the point is that we think it would've been really a lot harder to interpret what

these variables meant. If we wanted to say what he was doing at a given time, we'd have to save these 10 numbers instead of just like this nice one.

[0:28:49.6] SC: Right. That's the kind of thing we typically see in image interpretation, like in this image there is a umbrella with whatever probability and a girl with whatever probability. You've got this continuous probability distribution of the things that are in the image and you could do similar with what the video clip, the mouse is doing X, Y, Z with some probability and eating with some probability and grooming with another probability.

[0:29:17.5] DD: Right. The probabilities are continuous. There's still a big difference between having a model where the variables are continuous, and we're certain about them or where the variables are discreet and we're uncertain about them. This raises an interesting point which is that when you look at reality, reality is always continuous so why do we even have this discrete structure?

One reason that it devices naturally is because when we have to describe stuff to each other, in language we have to choose which were – is great like, "Am I hungry, or am I sleepy?" We don't have a whole –

[0:29:48.6] SC: A whole number.

[0:29:49.4] DD: Yeah. We can modify those with verbs. But again, we don't have some continuous signal that we can send to each other to – we can't just give each other high-dimensional vectors. That would be amazing. Then language of learning would be a lot easier. In fact, there's been some work recently by Open AI and some other groups on how to teach agents to communicate. They come across this exact same problem which is the agents have to choose a discrete word to say to the other agent. There's no way to back up and do that. There's no gradient signal that says, "Oh, you should've said this a little bit less." It's not fair what that even would mean.

[0:30:21.9] SC: Have you seen the movie Her by the way?

[0:30:23.6] DD: Yeah, I love that movie.

[0:30:24.7] SC: People have been telling me forever to watch it and I just started watching it on this trip. I'm not all away through with it, but there was this one part where for those who haven't seen it – maybe I shouldn't give any spoilers on the podcast. That wouldn't be right. There is an interesting point where this one AI talks about – is talking to a human and says, "Hey, can I go offline with this other AI and communicate post-verbally?" Which is exactly what you're talking about.

[0:30:52.2] DD: Yeah, exactly. It does raise a question. Why do we want these artificial agents to even use discrete words to talk to each other when they can just communicate post-verbally as you say? Right, and I think maybe the answer as well, we still want them to talk to us and they're going to have to probably use words to do that. That's one place where this comes up.

[0:31:10.4] SC: Okay. What are other things are you working on?

[0:31:12.8] DD: I'm working a little bit on meta learning. Just recently, I've decided that the way that I was approaching this and that I think is the mainstream weight now, I think there might be another way forward. There's been a lot of work recently where people have said, "Okay, I want to have a robot or a little agent that's going to be able to learn really quickly. I want to put it in a new environment and it's going to in a few seconds, figure out what's going on and then have a good policy how to act or something like that."

The brute force way to do this – no one really did very much until a couple years ago, was to back propagate through the entire learning procedure of the robot. Take those three seconds of him learning his way around the world. If everything he did is continuous, we can actually just ask if I had changed his learning rules a little bit, how much better would he have done on the task? This is fine. We can just compute this automatically with automatic differentiation.

This also shows up when we have to tune the hyper parameters of our model. We want to fit an entire neural network, but we have a learning rate, or like a regularization parameter that we said at once in the beginning and then it totally changes the outcome at the end.

A couple years ago, me and my colleague Dougal Maclaurin wrote a paper, where we actually did back propagate through the entire training procedure of training a neural network for hundreds of iterations. We got exact gradients and we could use gradient-based optimization to

tune our hyper-parameters. This is really exciting, because before that everyone had to use these black box methods, like random search or Bayesian optimization that is hairy.

Then after that, there was like these paper learning to learn by gradient descent by gradient descent. It's an amazing title. I wish that I had that as a title for our paper, which is like my foregoing title. Yeah, doing same sort of ideas and reinforcement with any people are really excited about this right now.

There's another way forward, which is to train a neural network to look at a problem. Maybe take in the hyper-parameters and then just directly output the optimal weights of a neural network. Skip the entire training procedure and just have that –

[0:33:21.9] SC: Not the hyper-parameters, the weights.

[0:33:23.6] DD: Yeah, exactly. Yeah. We're still going to have to tune some hyper-parameters, but we can train a neural network to just directly output the optimal weights. Okay, that sounds maybe dumb because if you think – well, if you think what I'm going to do is now train a whole bunch of neural networks like I did before and learn to predict the final outputs, then that would be slow and that would be a waste of time.

It turns out, we can train a neural network to produce a – I call this like a hyper network, because it produces the weights of another neural network. I can train a hyper neural network to produce an optimal neural network without ever having seen an optimal neural network. I can just have it start off produce a band neural network and then ask – use that prop to determine how should I have adjusted the parameters of my hyper-network so that it would've given me a better actual network. Then it gets very hard to talk about and everything is very beta and confusing.

[0:34:19.6] SC: I thought you were going to go in a direction of something like a GAN or something like that.

[0:34:24.9] DD: Maybe. The GAN does have this generator. I mean, I guess – yeah, I hadn't really thought about that. We could train a GAN to produce a network, but I guess the discriminate would have to see our train networks and real optimal networks and distinguish

between them. The whole point is I want to avoid ever having to start with a bunch of optimal networks. We can call this amortized optimization, where –

[0:34:51.1] SC: Amortized optimization. Interesting.

[0:34:52.6] DD: Yeah. Where we learn to do optimization by practicing and producing an optimal thing. This idea has been staring us in the face, because this is what variational auto-encoders do. They do amortized inference and that's like a name of this little sub-field where they say, we're going to learn to look at the data and train a neural network to produce the optimal posterior, like the optimal probability of the latent variables.

Again, these are trained in the same way where we never see the optimal posterior directly. We just, can use gradient signals to tell us how to get better and better. Then after we train for a while, we hope that we're almost optimal.

[0:35:30.6] SC: Cool. Well –

[0:35:32.6] DD: One thing I want to say is that –

[0:35:33.3] SC: Yeah, go ahead.

[0:35:33.9] DD: This second idea of the using the hyper network to avoid training is due to my student John Loraine. This is the beauty of this job is now, I get to look good because all these glowing students are coming up with ideas, and then I get asked about them. But I have to give credit where credit is due.

[0:35:48.4] SC: Awesome. Awesome. Well, thank you very much. This was a really interesting conversation and certainly has a lot of minorant firing trying to figure out all the stuff that we talked about. There is a lot of interesting stuff to dig into here. I appreciate you taking the time.

[0:36:03.2] DD: My pleasure.

[0:36:03.6] SC: Awesome. Thank you.

[END OF INTERVIEW]

[0:36:08.2] SC: All right everyone. That's our show for today. Thanks so much for listening and for your continued feedback and support. Thanks to your support, this podcast finished the year as a top 20 technology podcast on Apple Podcasts. My producers says that one of his goals this year is to crack the top 10, but we definitely cannot do that without your support.

What we need you to do is to head on over to your podcast app, rate the show, hopefully we've earned five stars, leave us a glowing review and share it with your friends, family, co-workers, Starbucks baristas, Uber drivers, everyone. Every review, rating and share goes a long way, so thank you so much in advance.

For more information on David or any or the topics covered in this episode, head on over to twimlai.com/talk/96.

Of course, we'd be delighted to hear from you either via a comment on the show notes page or via Twitter @twimlai

Thanks once again for listening, and catch you next time.

[END]

