EPISODE 600

[INTRODUCTION]

[0:00:00.3] JM: Applications of artificial intelligence are permeating our everyday lives. We notice it in small ways; improvements to speech recognition, better quality products being recommended to us, cheaper goods and services that have dropped in price because of more intelligent production. What can we quantitatively say about the rate at which artificial intelligence is improving? How fast are models advancing?

Do the different fields in artificial intelligence all advance together? Or are they improving separately from each other? In other words, if the accuracy of a speech recognition model doubles, does that mean that the accuracy of image recognition will also double? It's hard to know the answer to these questions. Machine learning models trained today can consume 300,000 times the amount of compute that could be consumed in 2012. That's a nice statistic, but it doesn't necessarily mean that models are 300,000 times better. These training algorithms could just be less efficient than yesterday's models. Therefore, they're consuming more compute.

We can observe from empirical data that models tend to get better with more compute. They also tend to get better with more data input. How much better do they get? Do they scale linearly with the amount of data, or the amount of compute? Well, that varies from application to application, it varies from speech recognition to language translation. We can't really say anything conclusively about all machine learning models improving, because of some specific metric, but models do seem to improve with more compute and more data.

Dario Amodei works at OpenAI, where he leads the AI safety team. In a post called AI and compute, Dario observed that the consumption of machine learning training runs is increasing exponentially, doubling every 3.5 months. In this episode, Dario discusses the implications of increased consumption of compute in the training process. Dario's focus is AI safety. AI safety encompasses both the prevention of accident and the prevention of deliberate malicious AI application.

Today, humans are dying in autonomous car crashes. It happens rarely; this is an accident. The reward functions of social networks are being exploited by botnets and fake salacious news. This is a malicious application of AI. The dangers of AI are already affecting our lives on these axes of accident and malice. There will be more accidents, there will be more malicious applications. The question is what to do about it? What are the general strategies that can be devised to improve AI safety? After Dario and I talked about the increased consumption of compute by training algorithms, we explore the implications of this increase for safety researchers.

I also want to quickly announce that we're looking for writers for Software Engineering Daily. We want to bring in new voices. We're focused on high-quality content about technology that will stand the test of time. AI safety is a good example of something that has not been written about much relative to how important it is. If you want to write, go to softwareengineeringdaily.com/write to find out more.

We're looking for part-time and full-time and volunteer contributors, people who just want to write about software engineering and people who want to turn it into a full-time job. We want to explain technical concepts and tell the untold stories of the software world. We just launched a new design at softwareengineeringdaily.com, so if you'd like to work with us, go to softwareengineeringdaily.com/write. You can also send me an e-mail directly, jeff@softwareengineeringdaily.com. I'd love to hear from you.

Let's get on with the show.

[SPONSOR MESSAGE]

[0:04:06.0] JM: Azure Container Service simplifies the deployment, management and operations of Kubernetes. Eliminate the complicated planning and deployment of fully orchestrated containerized applications with Kubernetes.

You can quickly provision clusters to be up and running in no time, while simplifying your monitoring and cluster management through auto upgrades and a built-in operations console.

Avoid being locked-in to any one vendor or resource. You can continue to work with the tools that you already know, so just helm and move applications to any Kubernetes deployment.

Integrate with your choice of container registry, including Azure container registry. Also, quickly and efficiently scale to maximize your resource utilization without having to take your applications offline. Isolate your application from infrastructure failures and transparently scale the underlying infrastructure to meet growing demands, all while increasing the security, reliability and availability of critical business workloads with Azure.

To learn more about Azure Container Service and other Azure services, as well as receive a free e-book by Brendan Burns, go to aka.ms/sedaily. Brendan Burns is the creator of Kubernetes and his e-book is about some of the distributed systems design lessons that he has learned building Kubernetes.

That e-book is available at aka.ms/sedaily.

[INTERVIEW]

[0:05:41.5] JM: Dario Amodei is the team lead for AI Safety at OpenAI. Dario, welcome to Software Engineering Daily.

[0:05:47.3] DA: Thanks.

[0:05:47.9] JM: You published a result recently that showed that the amount of compute used in AI training has been increasing dramatically. I'd like to discuss these findings with you. I'd like to start with just a little bit of basic discussion around AI to refresh people. Can you explain, what happened when an AI model is getting trained?

[0:06:09.8] DA: Right. We have two phases to at least today's AI models, they're a bit different from humans and that on humans these phases are more mixed together, although you can still see them as distinct. Let's take as an example computer vision model, right? That looks at a picture and classifies an object like a dog. There's a long period of training, during which you have to pass a bunch of training images through this model and update all of its internal

parameters. That's one computational process. It takes a long time. You often need millions of images to train on. You often need to train on each image multiple times.

Then at the end of the process, you can simply take the resulting neural net and then once it's trained, you can feed a single other image to it that maybe an image that wasn't in the train set and it says that's a dog, or that's a cat. The amount of computation that you put through a model at test time when you're classifying a final image is much less than what you have to put through to train it. Training requires all these infrastructure. Running the model is something that in some cases can be run on your phone.

[0:07:14.8] JM: Your paper is about the fact that this training process, which as you said is the compute-intensive portion of the two processes, at least today. The training process can use 300,000 times the amount of compute that was used in 2012. When you say compute, what is the resource that we're talking about more precisely? What does compute mean?

[0:07:39.1] DA: To get a little bit into details, we mean the number of arithmetic operations of whatever precision is appropriate for using the neural net that are performed in the full course of training. If train is spread across many machines and each machine is running a copy of the neural net, you would get this number by looking inside each machine looking at the number of arithmetic operations that gets done on the GPU on that machine throughout the course of training, adding that up across all the machines and that would be the number of flops.

Flops is of floating-point operations. It's a bit of a misnomer, because in some of the models the training isn't always – sometimes it’s single precision floats, sometimes it's half precision floats, sometimes it's integers, but we really just mean arithmetic operations, adds or multiplies within a neural net.

[0:08:25.4] JM: Why does the AI training process require so much compute?

[0:08:31.0] DA: There's differing opinions about whether this is just a fact about AI, or if this is a shortcoming of our algorithms. For whatever reason, it's the case right now that if you want to teach a neural net to do something like vision, or speech, or translation, or game playing, then

you need a huge amount of training data to allow it to do this. As I mentioned before, the image net data set for vision requires about a million images or so.

It's possible that once we get better at transferring to new tasks, or once we make progress on learning without supervision, that we'll be able to do this from a smaller number of data points. It's also possible that for the tasks we're working on, it just requires this many data points. Whichever of those two is the case, this is the situation we're in now, that it requires a huge amount of training data to build systems that work.

[0:09:25.6] JM: The result that the paper emphasizes is this dramatic increase in the amount of compute that is being used during the training process. Why is that relevant? Why is it relevant that we can consume so much more compute during a training run? Couldn't that just mean that well, the models that we're building today, we're just doing them less efficiently. We're building the models exactly the same, except we're doing it less efficiently and therefore, we're using more compute. How do we know that this increase, this dramatic increase in compute is correlated with an improved in something that is desirable?

[0:10:04.1] DA: Yeah. I think that's a really good question. I think the paper got mostly positive reception, but I think the minority of negative reception that we got was making a point similar to this. It's actually a point that I agree with, or at least think could be right. Here's a way I think about it. There are several – as the blog post says, there are several ingredients into making AI work, right? There's the algorithms, which if you don't have a good algorithm, you won't do well. There's the training data, which in the case of interactive environments is more like a game environment, but could also be static data. Then there's the hardware.

I would distinguish between a few things in the post. All the post is asserting as fact, is that one of these ingredients, the hardware is going completely crazy. I mean, if I was looking at I want to build something powerful knowing that one of the ingredients for that is going completely crazy, makes me wonder, I don't know how much of the other ingredients I need. I might need a little, I might need a lot, but looking at that trend, I would say, “Hey, I should pay attention, because it might turn out that I don't need as much of these other two ingredients as I think I do. Maybe I need a lot, and so I'm over-feeding this one ingredient, and so it really doesn't make a

difference, but we don't know if we're in that world or if we're in the world where this is the ingredient we need the most of."

As someone who tries to be cautious and tries to anticipate the speed of progress, this is something that makes me want to be cautious. Now, I think it's true. Two other things are definitely true. One is that it happens all the time, that we learn to build models that use the same amount of compute and perform better, or that perform just as well with less compute. Sometimes this is cited as hey, so this means compute and hardware don't – this means compute and hardware don't matter.

That hasn't been my experience at all. My experience has been that often, we do discover innovative things by experimenting with small amounts of training compute, but once we do that we learned to scale those things up and we use the same amount of compute that we use before to do something much more impressive, then we get more compute and we could do something even more impressive. These ingredients tend to add to each other. Whenever I see someone discovering a way, as has been done in actually several points on that graph, to do what they could do before, but with 90% less compute than they could do before, in my mind that just adds to this trend, right? It's another ingredient that's driving things forward.

[0:12:38.7] JM: You cite a paper in this blog post called Deep Learning Scaling is Predictable Empirically. That's the name of the paper that you cite, Deep Learning Scaling is Predictable Empirically. I think the idea of this paper is that there are things in deep learning that we do not understand. That's just the matter of fact is we train these models and we don't know exactly how they work, but they do work and we can make other empirical observations.

We're making one empirical observation in the fact that this model is good at identifying a cat. What is it doing to identify that cat? Well, I can't actually tell you at a low-level, but I can tell you that it's improving in identifying the cat. Similarly, there are other empirical conclusions that we can draw based off of looking at these different domains in machine learning, things like machine translation, language modeling, image processing, speech recognition. Then we could try to make general statements about the relationships between the size of training sets, the amount of compute, the accuracy of the model. Describe the significance of this paper, Deep Learning Scaling is Predictable Empirically.

[0:13:51.9] DA: A little background on that paper; so before I was at OpenAI, one of the places I worked was a Andrew Ng's lab at Baidu, which was very active in speech recognition which I which I worked on. One of the things we did in the paper that we released shortly before I left Baidu was we had this speech model in English and Chinese, and we did this analysis on it where we said, "Well, what happened? We have this huge amount of data that we're training this model on. What if we only take 50% of the data? What if we only take 25% of the data? What if we only take 10% of the data? What if we do it with the full-size model slightly smaller model?"

You can make these graphs that show how this model does is a function of these different inputs that you give it; data, computation, number of – amount of capacity in the model. We found these very smooth graphs and it was just a random observation that we made as part of the paper. Then I left. Then I left Baidu, but in the year or two after that, I found out that people had really followed up on this. For many different models, we're starting to make these observations that if you make a log-log plot of how much data are you feeding the model versus what its performance is, the plot has this very smooth shape, which is an interpretation is something like, every time you double the amount of data, the accuracy, the error rate of your speech model decreases by 12% or something like that, right?

It's a different scaling law for every system that we've observed. Eventually, of course you scale to it to – you do scale to a point, where it levels off, because the data you're training on has its own errors. Maybe the model was limited in some ways, but I was really struck both in the early work I did on this and in the paper that Baidu published later by how smooth these scaling laws are. That's one thing that makes me think, at least within a domain, right? This is a very important caveat. Within a domain, speech recognition, or vision recognition, within the domain and up to a point, there's this very smooth dependence on how much data you're using and how much compute you decide to use.

Again, even within the domain this eventually levels off and maybe you need a bigger data set, or a better model, or something like this. I think we may in fact be getting to the point in things like speech, image recognition, where we are starting to level off and we're at least on some axes have exceeded human performance. Although on other axes, we haven't. It is true that this

is only an observation within domains, but at the same time, I think there are other domains and things like reinforcement learning, or plain very difficult video games might be cases where we're still on the part of the curve where we haven't leveled off yet.

I've seen a lot of evidence at OpenAI and elsewhere that this is the case. My picture of it is okay, with in domains there's this smooth scaling up to a point. You need some innovation and models, but every time we have a new domain, we start we off at the beginning of this curve, and having a whole bunch of hardware available means that whenever we find a new domain, a new thing we can learn, a new data set we can train on, a new environment we can train reinforcement learning agents on, we're starting off with this huge amount of compute, and every time it increases, we increased down this scaling curve.

That makes me feel that the other ingredients like data and algorithms in many cases, we have enough of them that we can continue to make progress. By no stretch of the imagination is this always the case. There's going to need to be a lot of algorithmic innovation to solve all kinds of problems that we haven't solved before. We don't know exactly how much of it there needs to be and we don't know in how many domains. We don't know how far pure scaling can take us.

[0:17:48.1] JM: These two conclusions separately, so the first that your blog post was covering, the idea that you can observational see that 300,000 times the amount of compute is going into models today, than was going into models in 2012. The other observation that this other paper, this deep learning scalable predictability that is empirically observed, this other paper observes that within given domains, you can see that there is a predictable, empirically observable increase in model quality, or scalability to find how you want it to be defined as you add in more data, or more compute.

If you take these two discoveries together, this increase in compute that a model can gobble up and the fact that scaling seems to be predictable empirically, what conclusion can we draw from these two trends intersecting?

[0:18:54.8] DA: Yeah. I do want to make the caveat that I think the first trend, the growth and hardware which the blog post mostly focus on, I mean, that's something that you can quibble with the measurements a little bit, but we're asserting that as factual. The paper we link to and

that we write a couple sentences in support of, I would more say that's a conjecture that has some evidence behind it. It's only been tested for some domains. It may differ for different domains. I would more say that it's like – it's the tantalizing beginnings of seeing the science of deep learning and how it scales.

The blog post doesn't mostly focus on that second part, and I'm a little uncomfortable fully asserting it as factual. I would more say that we're seeing signs in that direction that you can make a case for it. Yeah, with that caveat, I mean, I think if it were the case that both trends were true, it would mean for at least a lot of the domains that we currently operate in again, vision, speech, image – vision, speech, translation, that we can get very far just by scaling things. We'll have the hardware to scale and the use of that hardware will actually translate into improved performance.

Now there is this tailing off, where it's possible, I don't know what happens within production systems at Google, or at other companies. It's possible that internally, we've already taken these systems to the point where they're leveling off and we may need some amount of additional algorithmic innovation. The models may not be right, or we may just have solved the problem depending on the domain.

I think the most interesting implication to me is if those two things are both true across a wide variety of domains, they could also be true across future domains. It could be that we have this crank where we discover a new problem. Maybe there's just one or two algorithmic innovations, where it's like, here's a new domain we can study, here's an algorithm for using it. If we have this crank of applying a lot of hardware to it and this other crank of applying a lot of hardware turns into good performance, then we might have a machine that allows us to very quickly conquer new domains that we might have thought would take a long time to solve.

If that's the case, then we're very quickly going to see AI capabilities that we don't have today. We can't do this at all to, "Oh, here's an idea for doing it," to we apply to but we applied a bunch of compute to it, there was some algorithmic innovation and two years later we can totally solve this task.

That's totally different from the plane systems in the world, but at least from a research point of view, I think this leads to the possibility of a world of very rapid and unpredictable progress, right? Where just because you can look at something that we can't do it all today, right? Something that we can't do it all today, like machines can't learn to recognize a new image the first time, a new class of image the first time they've seen it. Just because we can't do that at all today, doesn't mean that we won't be great at it two or three years from now. It might also take us 10 years or a 100 years, but I think the fact that one, maybe two of the three ingredients are right there for us means, that it's deceptive to say, we can't do this at all. Therefore, there's so much that – therefore, AI will never do these tasks, or it'll be a long time.

I think that unpredictability, it's both a source of excitement and a source of worry, right? Because society will need to adapt to these new things that we're already learning to do very fast and may start to learn to do even faster.

[SPONSOR MESSAGE]

[0:22:44.3] JM: Every team has its own software and every team has specific questions about that internal software. Stack Overflow for Teams is a private secure home for your teams' questions and answers. No more digging through stale wiki's and lost e-mails. Give your team back the time it needs to build better products.

Your engineering team already knows and loves Stack Overflow. They don't need another tool that they won't use. Get everything that 50 million people already love about Stack Overflow in a private secure environment with Stack Overflow for Teams. Try it today with your first 14 days free. Go to s.tk/daily.

Stack Overflow for Teams gives your team the answers they need to be productive, with the same interface that Stack Overflow users are familiar with. Go to s.tk/daily to try it today with your first 14 days free. Thank You Stack Overflow for Teams.

[INTERVIEW CONTINUED]

[0:23:57.7] JM: I do want to discuss the questions of safety and societal adaptation a little bit later. To talk at a more technical level continuing your points, in the last few years we have seen deployment of some specific algorithms that have allowed for better parallelism. Parallelism in the context of machine learning, at least as I've heard it. We did a show a while ago with somebody from Intel talking about data parallelism and model parallelism. Describe some of the techniques for parallelism in deep learning and how that applies to the quality of the models that we end up developing.

[0:24:40.4] DA: Yup. Yup. I think actually parallelism, which is one of the big things that's that's driving the trend that I showed in that blog post, it's been one of the big things that's been a limitation for deep learning for a while and is only gradually being lifted. Probably the best way to explain it is let's take let's take a particular model and let's return to the image recognition model as an example.

The idea is your training this image recognition model on a data set that has a million images or so. If you were completely naïve, one way you could train is you could pass an image through it, have it learned from the image, then pass another image through it, have it learn from that image and then do that 1.2 million times, right? That would take such a long surreal time that you wouldn't even be able to do it; you'd have to run it on one CPU and the number of serial operations you'd have to do, it would take you years to do it.

In practice, we both use GPUs to run a larger batch size; so I have my model, I have its parameters and I run a 100 images on a given GPU through that model, all with the same parameters and then I do a batch update of the parameters where I where I learn from all the images. I can also take this further, where I can run it on a few tens of GPUs. Over the last couple years, we've pushed it to maybe 100s and 100s and 1,000s of GPUs, where I have a large batch of something like a 1,000 to 10,000 images and I learned from those all at once.

The problem is, and if I go too far with this, the learning becomes inefficient and basically the extra parallelism is wasted. A good analogy to think about that is let's say I'm trying to figure out a new environment, like I'm trying to learn to play tennis or something, having feedback is really important, so I need to try something. I need to say how well it works and what my errors are and then I need to learn from that to try the next thing.

If I'm trying 10,000 things at once before I get feedback on any of those things, then for a while it's like, “It's great. I'm getting more experience at once. I can learn from it.” If I'm parallelizing things too much, then I'm trying all these things at random without actually getting feedback from them. Then when I try the next thing, I have to try a huge number, a huge number of other things.

The technical term is like slippage in your gradient updates, where because you're trying to learn from too many things at once without learning from the first things from the first parts of the batch without learning from the later parts of the batch, that the learning becomes inefficient. It was the case back in 2012 to 2014, or so, that you could use three GPUs to train a model in a few days, but you couldn't use a thousand GPUs to train the same model in a few minutes. You just couldn't parallelize the thing. That limited how powerful a single model could be, because how powerful a single model could be was tied to basically how much training resources you could pour into that single model.

[0:27:46.1] JM: If we're talking about domains like image recognition, or language translation, the problem is very well-defined ultimately. You want to identify an image, you want to translate a language. There's some subjectivity there, but generally the problem is quite well-defined. It seems like there are a lot of other domains where the problem is less well-defined. Even in something like world – developing an AI to play World of Warcraft, or to play Warcraft, or to play Starcraft, or to play an old arcade game, at least you can give it a reward function, like maximize the score, or maximize these different things across some trade-off function.

Things are pretty well defined, but if you talk about defining how a drone should fly, or how a car should drive around, it seems like there are so many other variables that it's a little bit harder to define – not a little bit harder, significantly harder to define what the problem that we're trying to solve is. Do anticipate any bottlenecks in the widespread applications of deep learning when we've exhausted these things that are a little bit easier to approach?

[0:29:07.4] DA: Yeah. I think that actually is one of the bottlenecks. I agree with you in the sense that the supervised learning tasks you described, they have a well-defined answer, even though the reinforcement learning tasks you describe, like game playing have a well-defined

answer. We still haven't gotten to the point where we can really turn the crank of compute and models on tasks that don't that have a particularly well-defined answer.

I would say there are a few places where we're starting to make progress on this. Generative models are an example of this, right? Where the setup here is I give you a bunch of pictures of things, like celebrity faces, or bedrooms, or cats or something, and you're asking the AI to generate more images that are drawn from the same distribution as those images. It will generate celebrity faces that don't correspond in the actual celebrity, but look plausible, look photorealistic.

I think we're starting to make – so we're starting to make a lot of progress on that. The other thing you mentioned about not knowing how to define success, not knowing what the objective or the goal is, that's actually been one of the subjects of research of my team at OpenAI. We did this paper called deep reinforcement learning from human preferences. The idea of it was exactly what you say. Like for many tasks, we would want an AI system to do, there's no well-defined score an Atari, or winning in Go, or winning in World of Warcraft.

What the human wants the AI system to do may be really complicated some, some complicated aesthetic thing, like doing a backflip that looks pretty, or rearranging furniture in a room or something like this. We developed a method for doing this, where basically the agent shows its behavior, examples of its behavior to a human, and the human rates examples of the behavior based on how close they are to what the human wants the agent to do.

Basically, the human is defining the reward function, the success criterion for the agent. Then the agent goes back, tries to do really well on that, then of course there's some subtle aspects of what the human wanted the agent to do that it doesn't get exactly right, so it shows more examples of its behavior to the human, the human discriminates between those. It says, “No, when you want to do this backfill you want it to be perfectly smooth. You don't want this little jerky part of the backflip.”

We did this only at small-scale so far, but the idea is that what I think we need is a lot of different ideas in this direction that could all be – we can try scaling all of them and see which one of them works. My guess is that eventually and maybe not too long, we will find one approach that

really does work at scale and really does get it on that log smooth curve of progress. It may be a little bumpy to find the right algorithm, but I think we're in the algorithm search stage to find the thing that works. I agree that's one of the most difficult things, that it looks we're getting very good at doing any task where we can quantify success, but we can't do tasks where we can't quantify success.

[0:32:27.0] JM: Your intuition is that we will figure out some method to approach this class of problem where it's harder to define what success is?

[0:32:38.8] DA: Yeah. I mean, I think the paper that me and my colleagues wrote about a year ago, Deep Reinforcement Learning from Human Preferences showed that you can't do this that small-scale. Then the question is just, does this work at large-scale for large tasks like driving, or doing ballet, or composing a song, or something like that?

[0:33:00.7] JM: One issue I'm curious about is dealing with large data sets. Even if we're just talking about these problems that are more approachable, like language translation. When I was at the TensorflowDev summit in 2017, I heard Jeff Dean say that they were not able to fit all of Google's available training data into the training process of a language translation model. I didn't really understand why that was. Why is it that you can hit a bottleneck with training data? Can you just keep getting more and more training data and just keep stuffing that into a model? What why do you hit a bottleneck?

[0:33:44.7] DA: I'm actually not 100% confident what Jeff was referring to. There are a few bottlenecks I can imagine. One is that you've just exhausted the amount of training data that Google can generate. I don't based on your quote. I don't think that's what he was referring to. Another possibility is that you can't feed your model fast enough, right? You have all these training data and you want to feed it to a model but, each image is like a megabyte or so, and if you want to train really fast, want to train a big model for a long time, each machine that trains the model has a limited input-output bandwidth. Maybe you can't train at the speed that you want to train to absorb the number of images you want to absorb, you exceed that training bandwidth.

It could also be a reference to parallelism, where even now you can only train on so many GPUs. If you can only train a 100 GPUs and each GPU can only process 500 images a second and you can only train the whole model for one month, or it takes too long, then then there's only a certain amount of data you can absorb. I think that further innovations in parallelism will I think continue to lift those limits.

Another comment is I think interactive environments may also save us here, where if you look at something like Atari or Go, or DOTA, which OpenAI is working on, basically what happens is the environment generates your training data, right? Your this DOTA agent, you're putting this environment, you take an action, then the environment generates the next state and you can just play the game over and over again and get repeated unique training data and you never have to store your training data. You just have to generate it on the fly, process it and then you can mostly throw it away.

Those may be ways around the bottleneck. Yeah. I mean, there are fundamental limits in how fast you can process data, how much data you can generate. I'm not sure which of those limits he was referring to.

[0:35:43.6] JM: Okay. I think we've done a pretty good job of setting up the premise of your paper, which is the fact that there is dramatic improvements in compute hardware that can be applied to deep learning, such that the training process of our deep learning model creation can consume compute dramatically faster. Whether or not that leads to better models, well that's not really in the scope of the paper; you talk about it a little bit. I think it's safe to say that programmers are pretty good at making use of additional compute when it's made available to them.

The fact that you can present large volumes of compute to the programmer and say, “Programmer, use this compute intelligently to generate better models,” seems like a driving force in creating better deep learning models. Of course, the amount of data that we have available to us is also improving. We have improvements in the quantity of data, we have improvements in the amount of compute, we have improvements in parallelism techniques. It seems like we are careening towards a world where models are very, very, very good. I mean,

we're in that world already. They're just going to get even better. You're the team lead for Safety at OpenAI.

I think that most people in the audience probably agree that AI safety is an important problem. Where they may disagree is the intensity of their belief on its importance. You wrote a paper called concrete problems in AI safety, which focuses on accidents. What I liked about this paper was that it is a clear-eyed view into some of the problems in AI safety, because I think it's very easy for people to make mistakes when they're thinking about AI safety, if they haven't thought about it deeply. Why are machine learning accidents a common concrete problem in safety?

[0:37:55.8] DA: Yeah. I would say a couple things on this. I mean, I think the fundamental source of it is when you run a machine learning system, you give up the guarantee that you have in most programming applications that you can at least in principle trace exactly what lines of code a system executed, why it did what it did. If I script something myself, I can at least say, I at least trace through the lines of code that are in principle understandable to me and say, "Okay, I understand why the system did what it did. Is this a bug? Is this an unanticipated situation?" I can I can track it down."

With a machine learning system, I have a training data set, or a training environment and I absorb specific examples in those training environments, but I'm then trying to generalize to other examples that are similar in some way to the training examples, but are not the same as any of the training examples. Every time a machine learning system is deployed, it's deployed in a situation that it's never seen before. The hope is that that's at least somewhat similar to the situations it was trained on, but sometimes it's not similar. When it's not similar, we really have on a theoretical basis, absolutely no guarantees about how that system is going to behave, right?

When I was at Baidu, in the early days of training our English speech system, we trained it mostly on American accented data, right? Then you would get this weird unbalanced speech system that would do great on anyone who spoke with an American accent. If you spoke with a British accent, or an Indian accent, or a Chinese accent, or a Eastern European accent, sometimes it would do really terribly, even though to a human these things don't sound all that different.

This basic unpredictability that I think also leads to a lot of the problems we're having with deploying self-driving cars, right? It's a lot easier to train in a simulator than it is to learn and train in the real world, but the real world is different from a simulator and we're always trying to make our simulators better, to make them more the real world. This shift between the training environment and a testing environment is something that's still with us. I think the concerning thing is often, with enough training and with enough iteration, we can make these things work in practice, but there's still this – we don't really have any guarantees, right? It's like, you do your best, you train. The more you train, the broader training environment you train in, the better you think you can do. We don't really have any guarantees on how the system will behave when it's deployed.

I think that's a matter of concern for me, right? I think the concern increases as the power of the system increases, where it's the concern is present when we talk about, kind of, today's systems, I mean, there were these infamous incidents of Google's photo system misclassifying African-American individuals as gorillas. It did very various horrifying things. If you go to reinforcement learning systems, then there's – with systems that are interacting with their environment, there's a new class of ways things can go wrong. This can be related to things like accidents and self-driving cars.

Then as systems do more and more of the high-level tasks that humans do, I think the danger will just go up, even as the basis of the worry. The conceptual foundations of it stay the same. That's mostly what my team works on, and the reason I do stuff like predicting compute is I'm just trying to understand how fast AI systems will get better, and if they're going to get better fast, or if there's a chance that they will, that's something we should know about.

[SPONSOR MESSAGE]

[0:41:56.3] JM: Kubernetes allows you to automate and thus increase the speed of deployment. But as you rapidly deliver, are you aware and prepared for issues in production? VictorOps Incident Management empowers progressive teams to ship to prod without worrying about a nightmare firefight. With VictorOps, your team has context to fully understand application and system health.

Coordinate on-call teams, collaborate when incidents occur and monitor Kubernetes through a large number of monitoring integrations; Datadog, Prometheus, and Grafana, and hundreds of other tools integrate directly with VictorOps to help you give deeper visibility into application health.

Visit victorops.com/sedaily to see how VictorOps can help you manage incidents and improve system observability. That's V-I-C-T-O-R-O-P-S.com/sedaily. Victorops.com/sedaily. See how VictorOps helps you build the future faster. Thank you, VictorOps.

[INTERVIEW CONTINUED]

[0:43:11.6] JM: To outline some of the classes of risks that you scrutinize in your paper about AI safety, the concrete problems in AI safety; safe exploration. An AI should be able to learn about a cliff that exists in a road environment without driving off of that cliff. You shouldn't have to drive off of a cliff in order to understand that cliffs are not something that you should explore if you're a car. An AI needs to be able to deal with new circumstances, so if a self-driving car sees a new type of situation it needs to be flexible enough to deal with it. If you have too brittle of a model, then it's not going to be able to deal with new circumstances.

You can have negative side effects. If you told a self-driving ambulance to quickly rescue somebody, it might drive on the sidewalk to get to that person and risk injuring pedestrians, which would be a negative side effect for sure. There's a number of other accidental side effects that you discuss in your paper. Are there any concrete solutions to these concrete problems in AI safety, or are you just outlining these and we're still at the very early days of being able to solve any of these things?

[0:44:31.2] DA: I mean, it's definitely true we're at the early days, but our team has already worked on a few ideas. We currently have in progress a project on safe exploration. You should see something out about that in a few months. We're thinking about how to anticipate problems that you might encounter during learning before they occur. Actually, one of the problems in this area, we mentioned scalable supervision and reward hacking. The paper I mentioned a few minutes ago, Deep Reinforcement Learning from Human Preferences was in part designed in

addition to addressing this issue of how do you define success, in part design to address this issue, because a lot of where things can go wrong, particularly systems that are reaching for a goal and interacting with an environment through a long series of events, a lot of how some of the safety issues can happen is if I'm specifying a goal and I don't necessarily know how a system is going to accomplish that goal.

It may end up doing it in a way that I didn't expect and that is harmful, because it's razor-focused on the goal I gave it, and there are all kinds of these other implicit things that it should or shouldn't do and it messes that up. The aspect of human feedback, where humans look at the behavior of the agent early on while it's training and say, “Is this the right thing to do? Is this the wrong thing to do?” Give feedback means that you have the ability for at least by the time the system is done training itself for its behaviors to reflect some complex notion of human preferences, human values, what the human actually wants the AI system to do.

That paper was our attempt to, in addition to attacking some problems within machine learning itself to attack some problems within the safety area. Yeah, we're definitely thinking a good deal about that. I think with safe exploration, there's a project in progress. We have one of the few people who's written deep learning papers on safe exploration working in our group. Surprisingly, there actually aren't – I mean, there are some, but there actually aren't that many up-to-date state-of-the-art papers in this area, which I think there should be a lot more.

Their colleagues at other institutions like at Google Brain, there's a good fellow who works on these things called adversarial examples, which are cases where by making a small change to the image presented to a neural net, you can make it classify things wrong. This both creates worries about systems being fragile, or unreliable and systems being attacked by malicious actors. That's a range of the stuff being done by me and by others.

[0:47:25.3] JM: I think it's great that you're thinking about this stuff, despite the fact that it's early days. Although my sense is that the way that this field is going to evolve is that we're simply going to restrict the deployment of machine learning models to domains where we do have a lot of control around the bounds of what the model could do. For example, we're going to have self-driving tractors that drive around farmland before we have self-driving cars that are driving around busy streets, because it's a more constrained environment. All you have is corn and

maybe a couple rabbits and maybe you have some cows in the area and you need to avoid the cows, but it's a more restrained area.

You can't have as many big problems. This would harken back to the same conversation we had earlier, where the models that we can solve for are highly constrained domains, like image recognition. Do you think that's an accurate way of looking at how things are going to evolve?

[0:48:34.2] DA: Yeah. A couple thoughts; I mean, I think there's definitely a fair amount of truth to that, right? Where if you look at self-driving cars, right? In 2010, we had reached the point where self-driving cars could make the right decision 98% of the time, or something, right? The original demos by Sebastian's run were from the point of view of is this thing driving pretty good? 98% is nowhere near good enough. We need them to be 99.999% in order to be better than humans at not killing people.

That phase, that deployment phase is going to take a long time and is going to involve a lot of yeah, a lot of operating in restricted domain. As that relates to safety, I'd say a couple things; one is that, in part, I see safety as advancing the field, right? Because if we can address some of these safety issues in a way that we can develop algorithms forward and we actually have confidence in them, that will allow us to deploy things in broader environments than we otherwise would have done so. Again, assuming that we've actually taken all the precautions and we're actually confident in them.

I see in some way safety as actively advancing the state of the field and the state of what we can deploy. The second point I would make is, I don't think this statement is universally true. I don't think that will fully restrict things to domains where we know what the system is doing, or where we know it can't do anything bad. I think that'll probably be true for things that are obviously safety critical, or life or death situations, like driving or medical diagnosis.

I think my bigger worry is about safety issues that are more subtle and aren't immediately seen as safety issues. I think we're going to end up building systems that do more and more on our behalf in the digital domain. We already do, right? We have systems that predict what we like and what we don't like on Facebook. We have all kinds of digital agents doing all kinds of things

for us. At some point, because we can we're going to aggregate and because it has benefits, we're going to aggregate those agents.

We're going to have big models that have some overarching understanding of what some large number of users are doing, right? Maybe even systems that manage power grids, make economic decisions, like systems that act on large complex connected systems. There, it's possible for something catastrophic to happen without us immediately telling that it's catastrophic, right? It's hard to – if I have a system that's like managing my Facebook account, or deciding what I like, there's no clean separation between the system is doing well-defined things on my behalf.

This system is helping Russia to influence the US election, or this system is causing me to get addicted to drugs on behalf of a pharmaceutical company. Right now these things are a bit metaphorical, but I think these things will become a lot more focused and technical issues as we automate more and more. I think the same could be true of military systems, systems that make financial decisions where it's not clear what's catastrophic and what's not. There's no fine – there's no bright line you can draw to put rails on what a system should do and shouldn't do, right? Sometimes system just should short sell a lot of stock. When should it do that, versus when is that a catastrophic decision, it's hard to know.

It seems to me that for those applications, we're going to be in a lot of danger and we're going to need to reach in and actually make sure that in a very abstract high-level way, these systems actually do act in-line with human values. If that doesn't happen, I think the amount of damage to society could, you know, I don't know when, but it could eventually be very high.

[0:52:31.5] JM: Yeah. You've written about this also, this – well, I mean, you're talking just now about catastrophic results, but those may or may not be delivered. They may be accidental, they may be deliberate, but what we can say with a lot of certainty is that there will be deliberate use of AI for creating new threats to global security. Describe how AI challenges global security.

[0:52:58.6] DA: Yeah. We've already seen early versions of it, again not involving AI yet, but what AI does is it allows things that were previously done by humans, or by simple scripts to be done cheaper, better and faster. All the rush of bots that tried to influence the election, actually I

don't know how actually effective they were compared to a world where they didn't exist. They definitely tried to influence the election.

A lot of those Twitter bots were pretty dumb, right? They were either humans in no content farms and Moldova or something, or they were very unsophisticated bots. As these things become more sophisticated, I think the scale of what can be done will go up and the set of actors that can do these things will go up. I see a lot of worries with things like drones. One worry is just old-fashioned hacking into self-driving cars. If that could be done at scale, that's something that presents a serious security risk.

I don't want to go into too much detail about how you can maliciously use AI, because I don't want to give anyone any ideas, but I think there's a lot in this area. I also think it's going to be quite inevitable that militaries are interested in AI to create autonomous weapons for intelligence, for counterintelligence, for surveillance. That presents a lot of threats to global security and global stability, right? It's new weapons, new ways of waging war, new things that could destabilize the balance of power between countries.

[0:54:33.2] JM: How should governments respond to this potential for deliberate malicious AI use?

[0:54:39.8] DA: Yeah, so that's a tough one. I think depends on the particular application. I'm conscious that I'm not myself a policy expert, so we have working closely with the safety team, but with different expertise. We have a policy team at OpenAI that thinks a lot about these issues. I should definitely say that I'm not the expert on what governments should do. I think broadly, I think extreme responses are bad. One extreme response is to try and regulate everything, right? I think that will be bad, because that would – there are many positive applications of AI.

If the United States regulates these technologies and other countries don't, then the bad effects still happen. These other countries, some of which may have more nefarious intent, will simply do the same thing. I think doing nothing at all is also an extreme response that I think is bad, right? As AI systems get more and more powerful, they're going to have greater and greater national security implications.

I'm also very uncomfortable the government doing nothing whatsoever. Specific things like what should the government do about autonomous weapons, what should they do about people hacking self-driving cars. I mean, I think that's a very case-by-case basis and my hope is maybe it's a forlorn hope that the regulations will be drawn by people who have a lot of knowledge and expertise in the area.

I think broad overreaching regulation of AI quote is a bad idea, just because there's so many different use of AI, right? We're like, how you regulate like an automated radiologist is going to be really different from how you regulate a self-driving car, is going to be really different from how you regulate a drone. Other than that I think we should avoid the extreme responses, I think it's really subtle and depends in a lot of case-by-case things.

[0:56:46.5] JM: To wrap-up, what are you working on at OpenAI now? I feel that we've touched on a lot of different areas of your domain. What are you working on today? How has these past results and past research led to whatever you're working on today?

[0:57:04.7] DA: I mean, the things I talked about in the last hour are a pretty good overview. I spend some of my time on this trend following and forecasting, just trying to understand where AI is going to be in two or three years and five years and 10 years, so that we can defend against the threats that seem most likely to actually occur. Some amount of my time is spent on that.

This work on learning from human preferences, we recently released a paper called AI safety via debate. This is an extension of the human preference work, where we try and make sure that agents are aligned with human interests by having them debate – training them to debate each other, try and convince a human of some fact where they're able to share some ground truth information. It's a little bit like the self-play that was used to train the DeepMind AlphaGo agent, except applied to trying to stay in-line with human values.

As I mentioned, we have this work on safe exploration. We're also starting to think about different directions in safety. Eventually, we may get into the adversarial example game. A lot of my own time has been spent on just growing the safety team. The team I run at OpenAI now

has about seven or eight people, depending on how you count. I think that's not enough, because I think we face the risk that AI could advance very fast. I want to grow that team as fast as I can.

Another thing I was involved in in the last couple months is I was really involved in writing OpenAI's organizational charter, which lays out some of what we're aiming to do in the long run, right? Particularly in the world, which I think we'll get to eventually. It could be soon, it could be a long time where we build AI systems that that match the human brain and human creativity and capability. OpenAI is founded on thinking about and preparing for that day, even if that day might be far away. Day to day, we just focus on advancing AI and defending against today's threats. Those are some of the things I've been doing the last couple months.

[0:59:18.8] JM: Is it part of the charter to make the cutting-edge of AI accessible to the public, or is it just to be internally at OpenAI on the cutting-edge?

[0:59:30.7] DA: Definitely the second and mostly the first. Yeah, there's a line in our charter about, we think that in order to work on basically the things I work on, like the societal impacts of AI and safety and policy and misuse, in order to do that credibly, we have to be as an institution on the cutting-edge of AI. What I do doesn't work without what everyone else at OpenAI does. The people who are working on robotics, who are training agents to play DOTA, who are making discoveries and generative models, that my credibility and frankly, our ability to get technical advice and collaborate with really good people comes from the fact that OpenAI is a leader in the field, and the set of things we're doing basically the recipe just wouldn't work without that. That's the second part.

Then then the first part is yeah, I mean, as indicated in the name, OpenAI has been pretty focused on releasing its results to the world and making them available. There is one addendum to that, which we mentioned the charter, which is that we think that there's a small fraction of research that may have safety, or security, or malicious use implications and particularly in the fast-progress world, that fraction may grow over the years ahead. We expressed this long-term intent that the field has generally been very open and published a lot and we think that's good for now, but we may at some point be headed into a new world where it makes sense to publish less, or discourse norms around publishing, maybe more like what they are in the security

world, or the cryptography world. We're excited to navigate that transition, where we can still get as many of the benefits of openness as we can, while also respecting our ethical obligation to not do harm.

[1:01:24.0] JM: All right. Well, Dario you've been very generous with your time. I am very thankful for you coming on Software Engineering Daily. Thanks for writing such a great paper. It was thought-provoking and I enjoyed talking to you.

[1:01:35.0] DA: Thanks for inviting me.

[END OF INTERVIEW]

[1:01:39.5] JM: GoCD is a continuous delivery tool created by ThoughtWorks. It's open source and free to use and GoCD has all the features you need for continuous delivery. Model your deployment pipelines without installing any plugins. Use the value stream map to visualize your end-to-end workflow. If you use Kubernetes, GoCD is a natural fit to add continuous delivery to your project.

With GoCD running on Kubernetes, you define your build workflow and let GoCD provision and scale your infrastructure on the fly. GoCD agents use Kubernetes to scale as needed. Check out gocd.org/sedaily and learn about how you can get started. GoCD was built with the learnings of the ThoughtWorks engineering team, who have talked about building the product in previous episodes of Software Engineering Daily, and it's great to see the continued progress on GoCD with the new Kubernetes integrations.

You can check it out for yourself at gocd.org/sedaily. Thank you so much to ThoughtWorks for being a long-time sponsor of Software Engineering Daily. We're proud to have ThoughtWorks and GoCD as sponsors of the show.

[END]