DAISY THE GREAT:	00:03	Music
CRAIG:	00:07	Hi, this is Craig Smith with a new podcast about artificial intelligence. Every other week I talk to people who are making a difference in the space. And this week I continue on the theme of thinking robots with Sergey Levine who works with my previous guest, Pieter Abbeel, at Berkeley's AI research lab. Sergey focuses on machine learning for decision making and control. And we talked about developing a robot sense of touch and about robot dreams and Sergey's dream of robots operating autonomously in the real world. I hope you find this conversation as interesting as I did.
DAISY THE GREAT:	00:45	Music
CRAIG:	00:50	Can we start by you telling us how you got to where you are today - where you grew up and what your interests were as a child and then through high school?
SERGEY:	01:00	So I'm originally from Russia. I was born in Moscow. You know, mostly I was educated in the United States. So, I've been in the US since I was about 15 years old. I started college to work on computer graphics, so movies, video games, things like that. And part of what really drew me to that is that it seemed like it was possible to use computers to essentially create like artificial environments. As a kind of a facsimile of the real world with whatever aspect you want to change. Whatever aspect you find imperfect or whatever aspect of the world you might change to be more interesting. You kind of create these worlds, these environments.
SERGEY:	01:33	And in computer graphics, probably the most challenging thing to create in virtual environments, whether you're doing film or video games, is the people that populate those environments. Right? So, we've come a long way in terms of being able to create realistic lighting, realistic material properties, appearance, et cetera, but the actual behavior, the thing that demands intelligence, that's the part that's hard.
SERGEY:	01:54	So when I started my PhD at Stanford, the thing that I want to work on was better techniques for animating virtual characters. And it turns out that better techniques for animating virtual characters means better artificial intelligence because that's the thing that the virtual characters were lacking. And so, I worked on that for a while and then kind of towards the end of my PhD, what I came to realize is that a lot of the methods that I'd been developing to try to create more intelligent, more lifelike virtual characters were not really specific to virtual characters at all. They were just things that, you know, emulated some basic level of intelligence at a very low level in terms of like motor control.

Can you walk and you run, can you jump, et cetera. And can you do so in a way that reacts with some degree of common sense to obstacles, you know, obstructions and so on.

SERGEY: 02:33 So, after my PhD I thought that, well maybe it would be good idea to see how some of this stuff actually works in the real world. Maybe it's not just for video games and films, but it can be used for real physical systems. So, during my postdoc, which actually was here with Professor Abbeel, that's what I worked on. I actually took some of the methods that I developed during my PhD, which at the time I'd used for character animation or graphics and try to see if I could use them to get that robot out there to actually learn to perform some basic manipulation tasks. And there's perhaps a certain hypothesis, more accurately call it a guess, that I have that I feel like I've always been kind of operating under with this work that if we begin with very basic behaviors that to us are very easy, very straight forward to do like doing basic things with our hands, walking, running, et cetera.

SERGEY: 03:16 Then that can act as a very good stepping stone towards more complex behaviors that we would consider sort of emblematic of intelligence. So instead of starting with getting the computer to do some very cognitively demanding task, like playing chess or something, maybe we start with the things that are very easy to us and very hard for the computer and build our way up from there. And that's why I really wanted to work on, when I started working on robotics, on basic manipulation skills. So, can the robot, you know, take a lid and put the lid on something. Can it put together some Lego blocks, things like that. Things that like, you know, two to three-year-old children do, not necessarily, you know, world class professional players of some difficult game. And part of why I thought that this would be a good idea is, and this is going to be a very vague kind of notion, it's not a uh, rich or scientific hypothesis, but I'll say it anyway.

SERGEY: 03:59 In the field of computer vision, the field of computer vision was one of the first to really be revolutionized by deep learning, one of the things we saw, there kind of in the early days circa around 2010, is that when people switch to using deep learning techniques as opposed to manually engineering the features. So, the way computer vision used to work was someone would sit down and say, well what matters for detecting cars? Well, maybe cars have edges, they have an outline, they have a silhouette, they have wheels. So, I can detect circles, detect outlines and then if the outline looks like this and the circles are over here, I'm going to say that it's a car. So, you kind of designed these low-level primitives and then maybe there's a little bit of learning on top of those primitives that sort of puts them together. With deep learning, the thing that changed is that those low-level things - the edge as the circles, et Cetera - would emerge automatically from the data.

SERGEY: 04:34

And when people looked at those at the actual features that were learned by deep nets, when they were trained directly on the pixels in the image, they saw that the features that were emerging, well first they looked very natural. They looked like edges and corners and things like that. Things that to us seem like very reasonable feature. And they actually matched what people saw in the visual cortex. So, if you actually analyze the features that the visual cortex is sensitive to in primates, you'll find features that look very similar to what the early layers in convolutional neural networks pick up on. And then people said, well, okay, we can analyze what happens in higher levels of the network. We can't do that in the brain because it's too complicated.

SERGEY: 05:11

But you can do it in a convolutional network, and they saw that higher up in the network, just as you might expect, higher level features start to emerge, compositions of composition of edges, little parts. You know, if you have trained these things on faces, you get noses and eyes and so on. So, I thought, well, maybe the same thing would hold for behavior. Perhaps if we figure out putting the lid on the bottle, some other basic motor control primitives, if we can learn that from scratch, then the same basic technique would then be able to compose those motor control features into higher level motor control primitives. And you know, in the end motor control is everything that people do. Your brain only exists to move your body. All the change you make in the world is through your body. So, if you get to a high enough level, eventually you'll be able to do anything in principle that a person could do just by sort of building up.

CRAIG: 05:48

Yeah. And why work on physical robots as opposed to virtual? You know, I did an episode and spent some time in New York with Julian Togelius, who talks about sort of similar goals using AI to manipulate the real world, but, as he says, robots you have got to, you know, keep fixing them and keeping the oil in and all that stuff. They operate at a much slower pace. So why physical?

SERGEY: 06:12

Yeah, that's a very good question. It's a question that I get a lot. It's a subtle question to answer, but maybe one way that we can discuss this is the following: that intelligence doesn't just emerge by virtue of having the right algorithm or even the right brain. It emerges through the interaction of the thing that is intelligent, whether it's a brain or a computer, and its environment. And the environment is actually indispensable, especially if your intelligent machine is based on learning. I think it's at this point fairly widely accepted that a lot of the intelligence that people have is a consequence of learning, maybe not all of it, but a lot of it. And that learning is mediated by our environment. If we're in an environment that doesn't provide us with stimuli that doesn't actually force us to learn anything, then we will probably be much less intelligent.

SERGEY:	06:52	Our society recognized this a long time ago. That's why we have schools and colleges and so on, to put young humans into environments where they are stimulated enough to become intelligent. I think that the role of the environment is actually paramount. So, while I think it is possible in principle for us to develop simulated environments that kind of demand intelligence in the same way that the real world does. And that's actually what I spent a lot of my PhD in college days doing. I think it's actually very hard. I think it's hard because there are a lot of unknowns there. We don't actually know what in particular causes this kind of need for intelligence. And I think it's not even as simple as saying, well let's design a very difficult simulator. You can design a very difficult task like playing a very difficult game, for example.
SERGEY:	07:31	That task might demand intelligence in order to win in order to do it properly, but it might not provide you with the kind of affordances that you need to acquire that intelligence in the first place. So, if I just give you, for example, a very difficult math test, then it's maybe reasonable to suppose that you need a very good knowledge of math to pass the test. But that test alone might not be enough for you to acquire that knowledge to acquire that understanding. So, I think the problem of building the kind of environments that will result in a learning machine acquiring something that we might call intelligence is actually itself a very, very difficult problem. And by operating directly in the real world, we can side step, we cheat a little bit. We say, well let's just remove that unknown and let's just worry about the learning machine because the learning machine will be in an environment that we already know is sufficient for the emergence of intelligence, intelligence, which is our own environment.
CRAIG:	08:09	And so you had this interest initially from animation. Where did the interest in animation start? Were you a gamer or ...
SERGEY:	08:16	Oh yeah. I think that for me, for basically, you know, through college and throughout my PhD that games and movies were really the thing that was incredibly fascinated by. And I think the reason I was fascinated by it, it was one of the few ways that a person could essentially reshape a world in whatever way they pleased. If you can design a video game or if you can make a movie or something, you can, you can essentially create to a very high degree of fidelity, anything you could imagine - almost an entirely new universe. To me that was very fascinating.

CRAIG:	08:42	And you were born in Russia, you said you came over at 14?
SERGEY:	08:45	I actually lived in Turkey for three years and then came to the United States.
CRAIG:	08:48	Oh. And are your parents academics?
SERGEY:	08:50	Yes. So, my parents, they are both computer scientists. They actually worked on the space program in the Soviet Union. Then they moved to Turkey to continue their work on formal verification and now they work at Microsoft.
CRAIG:	09:00	Oh, and your childhood and high school years, you were aware and involved in computer science or what was happening in software development and that sort of thing?
SERGEY:	09:11	Well, I was involved in what was happening in video game development.
CRAIG:	09:15	Yeah, yeah. But it's not, Rich Sutton, for example, you know, his dad was an executive. There was no ...
SERGEY:	09:20	Sure. Yeah. Well, and in that sense, I do think that I benefited tremendously from the fact that my family was very active in terms of engineering and certainly I got tremendous support from both of my parents as far as learning computer science, learning the mathematical foundations of all this stuff. And that was, I think, enormously helpful for me to get to where I am now.
CRAIG:	09:36	Did you do any of this in high school?
SERGEY:		I did amateur game development in high school.
CRAIG:	09:41	Yeah, but you weren't programming and that sort of thing?
SERGEY:	09:44	There was a little, but I don't think I really worked on artificial intelligence until my PhD.
CRAIG:	09:56	Could you sort of take us through how the artificial intelligence or machine learning in controlling robots has developed? Just very briefly, the history of that.
SERGEY:	10:05	It's a complicated history because in contrast to a lot of other fields where machine learning has been applied, it's very difficult to sort of scaffold the robotics problem in a way so as to isolate a particular dimension of it. So, robotics is very fundamentally an integrative science, which means that you need to get all the pieces to work in order for it to do something. So, if I'm studying, let's say computer vision, I don't

have to really worry about building a better camera, for instance. I just get some pictures and I'd do something with those pictures. But with a robot you kind of have to worry about the whole package.

SERGEY: 10:31 And as a consequence of that, the intersection of robotics and machine learning has historically I think been almost like a little bit of an uneasy truce because if at some fundamental level things really have to work for real in the real world and you have a component that is not yet ready, which for most of its history, machine learning has been not yet ready, then you want to kind of isolate it and put it in its own place on sort of support it with as much as possible of the existing reliable generally manually designed machinery.

SERGEY: 11:00 So, robotics has up until now I would say been to a very large degree dominated by mature, well developed controls techniques that are based on a fairly kind of classical model-based approach of decision making. So, the classical model-based approach to decision making means that you characterize the physical behavior of your system. You write down some equations that describe how it will work. You essentially do a bit of Algebra and set the right hand of the equality sign to be what you want the machine to do, solve for the left-hand side, get some settings for the controls that will make the left side equal to the right side. Then run that in the real world. And that's very tricky because it requires you to actually have all those equations and the equations have to be essentially perfect. They have to accurately describe what really happens in reality. That's a very hard thing to do.

SERGEY: 11:41 So, in principle machine learning can help us fix that. But, historically it's been a very long and difficult road partly because kind of marrying machine learning techniques with the kind of analytic model-based controls approaches is pretty non-trivial, like they almost speak different languages. Machine learning speaks the language of probability, whereas controls speaks the language of calculus and Algebra and while you can put them together, it's quite tricky. You could simply discard one of them. You could say, well let's take a pure learning-based approach. But then you run into that problem that robotics really is integrative and the whole thing really needs to work directly in the real world on the whole complete robot.

SERGEY: 12:14 And I think for that reason we haven't really seen very much machine learning used in actual robots that are out there in the real world until very, very recently. Very recently, meaning in the last three or four years, certainly in the area of robotic perception, learning based components have really come to the forefront. So, if you have a robotic perception system, it is using learning because there's essentially no other way to do it. But direct robotic control, we're basically right there on the, on the cusp of it right now that probably right around now in 2019 is the inflection point where robotic control goes from using the tried and true analytic model-based methods to more and more learning-based techniques.

CRAIG: 12:47 So the robots we've seen in industry are all using this model-based approach, not machine learning.

SERGEY: 12:52 If you see a robot in a factory practically guaranteed. Yes. Yeah. If you see an autonomous car, unless it's a very small and very ambitious startup, practically guaranteed that it's using a classic planning approach that has its roots along long time ago.

CRAIG: 13:11 So this touch paper that you did with the gelled tips on the robot digits, has there been a lot of work on this touch-based approach?

SERGEY: 13:23 This question is actually a very good fit with the previous question because in touch sensing, touch sensing I think illustrates this kind of relationship between classic model-based controls methods and learning based methods very nicely. Touch is a very poor fit I would say, and some of my colleagues might disagree with me, but I think it's a very poor fit for classic model-based controls methods because you have to write down those equations. You have to characterize the system. In the way that a touch sensor works ends up being very difficult to characterize. In fact, a lot of the really hard work that people have put into designing touch sensors really deals less with the problem of how do you get a touch signal out of it and more of the problem of how do you calibrate it, how do you convert it into a quantity in Newtons or in pounds per square foot or whatever you want that actually you can plug into a physics formula. And that part ends up being hard.

SERGEY: 14:06 So it's not difficult to build a little device that will respond with an electrical current to changes in pressure. What's difficult is to build a device that is calibrated that you can plug into a formula. And learning allows us to not have to do that. So, I think that actually we might see widespread adoption of learning based techniques for robots that require touch sensing for much the same reason that we saw widespread adoption of learning based techniques for vision. Because these sensory modalities are textbook physics formulas and might give a crude picture of what they're doing. They don't really follow the rules perfectly because there's just so much complexity in the world. And if you don't have to fit that touch signal into a clean physics formula in terms of Newtons and pounds per square foot and whatever, but can just plug it into a learning machine that just spits out the answer that you want, you can get away with much less work in terms of actually building the sensor.

SERGEY:	14:49	So the particular sensor we used in that project. It was actually developed at MIT by Prof. Ted Adelson's group in his students and they did a very good job of actually taking this basic design and figuring out how to calibrate it, how to get, you know, 3D geometry out of it and so on. We actually used a much cruder design based on that work, but we just didn't do it as well. We 3D printed all the parts. We cast our mold in a haphazard, not entirely well thought out way and our sensor was objectively much worse than theirs. But all we did is we just took the images just coming out of the camera and we plugged it into a big neural net and we didn't need to know what the force was in Newtons or anything like that. So, we didn't need all of that sophistication and we can get away with something much cruder.
SERGEY:	15:26	The way the basic method worked was that it learned how to predict. So, it learned how to take the current reading from the touch sensor, a sequence of actions that the robot is considering executing and predict what the touch sensor will perceive if you execute those actions. So, if I have my finger on the edge of this cup and I'm thinking, well what will happen if I move my finger towards me, maybe I'll feel the curvature of the cup. If I move it the other way I'll feel the curvature in the opposite direction. So that's essentially the kind of prediction that it was making. And then I could do basic manipulation using the touch sensor. It could put its finger on a 20-sided die and it could imagine that if it moved the finger a little to the right, then the other face of the die would be facing it.
SERGEY:	16:01	If we're moving in the opposite direction, then it would be the opposite face. And then it could use this too manipulate the object to put it into a desire pose.
CRAIG:	16:07	And it could also predict how much pressure was needed based on the density or the weight of the object so that if it needed to pick it up.
SERGEY:	16:15	Well, so the interesting thing about these models is that they kind of do what's necessary to solve the task. So, we can try to analyze them. We can try to figure out are they predicting pressure or are they doing something else? It's a little bit of guesswork. We can sort of construct little experiments to try and measure that. But in the end, the answer is always something like, well, they're doing a little bit of everything. They're doing whatever they need to do in order to solve the task. An example, like, do you drive a car?

SERGEY:	16:38	Right? So, you've probably been driving a car for quite a while and I could ask you to describe how do you merge onto the highway? And you could say, well, okay, so I turned the wheel like this. I look a little bit to my left, I look a little bit to my right. I look in the rearview mirror, but you don't actually think about all those things when you doing it. Their all just kind of programmed into and you're kind of basically doing everything all at once. So, while you could decompose them into parts, it's really just kind of retrofitting some rationalization into a process that is much more complex and interwoven than your description would suggest, and I think it's the same way for the machines.
CRAIG:	17:06	Yeah. The machine learning system that you're feeding this touch sensor data into. Is this a convolutional neural net?
SERGEY:	17:13	So, for the particular touch sensing project that you asked about, the way the touch sensor actually works is it's a piece of gel with a camera embedded inside of it and the camera essentially sees indentations in the gel. So, the gel has an opaque coating. But when something presses against it, it can see the indentation on the underside of the gel. So that means that the data coming out of the sensor consists of camera images and we can plug it into a model that processes camera images. In fact, the model that we used was originally designed to do the same kind of prediction-based control from regular camera images. A previous experiment, we had a robot with a camera and it would push objects around on the table to try to rearrange them and it would learn to predict what it would see next from its camera. And if it can predict what it'll see next, it can take the actions for which the model predicts the outcome of the user command.
SERGEY:	17:54	So if the user asks me to move this cup over here, then I would take a picture of where I want it to be, put it back, ask the robot to do the task and the robot would try different actions in the model. See for which action does the model predict the cup will be in the right place and then go and execute that action in the world. The touch base model works exactly the same way only instead of camera images from a regular camera, it's images from that camera are embedded in the sensor. It's actually exactly the same model.,
CRAIG:	18:17	Right. Which is not from the layman's point of view what we think of as the sense of touch. Right. What you were talking about earlier, the more traditional model of measuring pressure and that sort of thing. Is this the future of touch in robots using this vision system or is there a way of applying that to the pounds per square inch data?

SERGEY:	18:38	I think that the real answer is that it probably doesn't really matter. I'm not a hardware designer. I don't really know which particular sensor people will use in the future, but to me the lesson from this work is that with a fairly general learning technique, it doesn't matter all that much how the sensor works. Whatever information the sensor produces, so long as there's some information in that stream of data from the sensor that allows you to infer the thing that you need to know. A good enough learning system will figure it out. So, whether the sensor produces those camera images or whether it produces pressures or whether it produces some conductivity reading or whatever kind of principle it works on. So long as whatever numbers come out of it can be used to figure out the information that you need, a sufficiently powerful, sufficiently general learning algorithm, will figure it out.
SERGEY:	19:16	And I think that has tremendous implications for how we're going to build robots in the future because it means that we don't have to worry so much about how exactly we're going to characterize our sensor or calibrate. So long as the sensor produces information that that correlates with the thing that we need, we can just quickly prototype a new kind of sensor, stick it on there, run the training, and if the machine works that means the sensor was good enough and if it doesn't work that means the sensor needs to be improved. I think it will have huge implications for how we design robots actually.
CRAIG:	19:40	Yeah. Again, from a layman's point of view, am I correct in that the previous, you know, if I want to pick up a ball in a factory, I have to calibrate the robot very precisely so that the gripper closes and stops closing at the precise dimensions in order to pick up the ball. Whereas with a touch sensor it would sense where it has to stop pinching.
SERGEY:	20:01	It could do that, but I think maybe more to the point it would just do whatever is necessary to succeed at the task. There's a little bit of a nuance there, which is that you have to define what success means, whether you're using a reinforcement learning technique or this predictive thing. It's all kind of the same, but you do need to define what you want it to do. There's something very subtle there because for kind of classic analytic control methods, we kind of bypass that because we need to specify the task in so much detail that there isn't really room for any ambiguity in the objective. You know the task for the robot is not to pick up a thing, it's to follow this particular very detailed trajectory that an engineer designed very carefully with like you know, days and days of work. For a learning-based technique, you need a higher-level objective like pick up the thing or maybe assemble the device or whatever it's supposed to be doing. But then you have to be careful that you actually

		define the objective that results in the behavior that you want to see. We've been actually studying this problem quite a bit. I could tell you a little bit about that.
CRAIG:	20:48	That would be fascinating.
SERGEY:	20:49	So one thing that we've been studying a fair bit is whether we can actually use computer vision systems to define objectives. So, you know that a computer vision system could do a good job of recognizing whether it's looking at a picture of one object or another object. Maybe it's looking at a picture of a bottle with a cap already on it, or a bottle without a cap on it. So now if you want to train a robot to put the cap on the bottle, you can just give it this classifier from computer vision and tell it well, make sure that whatever you do results in a scene where the classifier tells you that the cap is on the bottle. And that seems like a very reasonable idea, right? The trouble is that if you do that, the robot will cheat. It's like the genie, you know, it gives you three wishes and fulfills them, but never in quite the way that you want.
SERGEY:	21:25	And we've actually observed exactly this behavior. If you just use a standard classifier as your objective, the robot will force the classifier to output the right answer by doing something kind of weird that causes the classifier to see something that it's never seen before. So maybe it'll take its finger and it'll kind of shove the finger into the camera and position it just the right way so that the classifier gets messed up. And it'll say, oh now I'm winning, now I'm succeeding at my task. So, one of the things we've been working on quite a bit over the past year, is actually designing algorithms that can allow us to mitigate problem. It turns out that one simple thing you can do, it's going to sound a little silly, but it actually makes sense mathematically, is you can just have the robot attempt the task, but then you update your classifier with the additional data collected from the robots attempts and you just label all the data as incorrect.
SERGEY:	22:05	It's like I give you a test and whatever answer you're giving me until you know that answer is wrong. So, you have the example of the right answer from that one time that I showed you the bottle with a cap on it and everything else do you do. I tell you that's wrong. And it turns out that you can actually show mathematically that if you do this you will actually converge to the right behavior, that it will actually end up putting the cap on the bottle because whatever other crazy stuff it does, it's going to be told that it's wrong. So, we've been extending this line of work, we actually have a paper that will probably come out in a couple of weeks that shows this thing running on a real robot and it seems to be a step in the right direction. That said, the bigger kind of problem of objective specification is still a wide-

open problem. In more complex open world environments, there are many, many ways this could go wrong and I think it's an area that needs more work.

CRAIG: 22:42 And, and the labeling that something is wrong. Is that done manually?

SERGEY: We've done a little bit of both. So, the simple way to do it automatically.

SERGEY: 22:50 So you literally just say whatever the robot did, it's labeled as wrong. Sort of the worst possible test the robot could get turns out it still results in a convergence to the right objective. You still have that one initial example of the cap on the bottle problem. But you can actually make this go a lot faster if you do incorporate some human feedback. And one of the things we've done is we actually have an active learning system where the robot attempts the task and then if it's unsure about whether it did it right or not, it actually goes and asks a person to basically the label this particular outcome and that makes it go a lot faster.

CRAIG: 23:16 Faster meaning, I'm just curious, are you talking about hours, days, weeks?

SERGEY: This is all in terms of hours.

CRAIG: I don't know whether it was in the paper or whether it was in an article about the paper. One of the fascinating things I read was whoever was writing was talking about try picking up an object even though you have vision if you have no feeling in your fingers. And that to me sort of brought home how important touch is even for a robot because otherwise you have to specify very exactly what you want it to do.

SERGEY: 23:46 Yeah. We actually had that discussion in our paper on touch-based grasping, and there's a wonderful example from about 10 years ago of a study where they actually did exactly this with human subjects. They anesthetized the fingers of the subjects and they didn't have them grasping objects, they had them trying to strike a match and turns out that you basically can't do it. If you don't have the sense of touch in your fingers striking a match is essentially impossible for a person.

CRAIG: That's fascinating.

CRAIG: 24:15 This latest, I don't know if it's the later or earlier paper about reinforcement learning without the reward. Can you describe that a little bit?

SERGEY:		Yeah, so this is actually a nice transition from what I was saying earlier about specifying objectives. So, one way you could specify objectives is by telling the robot in advance what it needs to do and then maybe using some kind of additional machinery for [GARBLED]. Another thing you could do is you could just have the robot practice in advance without being told what it will be tested on essentially. And you can think of that practice as kind of a play phase. So, it's going to interact with the world, learn about how the world works, learn about how different sequences of actions lead to different outcomes. Use that to sort of build up its knowledge and then later on someone will ask it to do something and it can draw on that knowledge to figure out how to perform that task.
SERGEY:	24:55	Of course, saying that it's learning without any reward is a little bit of an oxymoron. Somebody needs to tell it what it's supposed to do. And in that case, what it is being asked to do is to essentially diversify its experience as much as possible. So, do as many different things as you possibly can. And that causes it to essentially explore its environment. So, the way that this method works, if you were to look at a robot actually executing it, the robot gets switched on and it starts doing things and it does one thing for a little while maybe plays with this object, then it goes and plays with something else, then it does something different and so on. And then just kind of cycles through all the possible things in this environment it can try playing with. What it's learning when it does this is it's learning how to reach particular outcomes.
SERGEY:	25:29	So that then once it's done learning, a person can come along and give it a particular goal. Right now, the goal is specified with a picture. So why a picture? Well, because it's just the simplest thing we could think of. Maybe in the future. It could be language or something else, but pictures seem simple. So, I show it a picture and the picture basically indicates I want the world to look like this. So, if it's in front of a door, I show it a picture of an open door, that means I want it to open the door. And then the robot uses what it learned from that play phase to go and do the task.
CRAIG:		How much transfer learning is taking place in that. You could have two systems working together to complete a task, but if you had one system, it retains the knowledge from one part of the task to apply to the other.
CRAIG:	26:04	Is that what's happening in this case? I mean, is there transfer learning?
SERGEY:		I think there's a two-part answer that I can give to this. The first part is where we see this going in the long run and the second

part is what it's doing right now, but as a quick spoiler, what it's doing right now is a very small part of what we hope to eventually see. So, our hope in the long run is that work like this can be a stepping stone towards a future where you have many networked robots that are just out there in the world and when they're not busy doing something more productive, they'll just play with their environment and learn. They'll essentially say, okay, if I'm not currently tasked with a job, if my human owner doesn't want me to do anything in particular, I'll just use my free time to practice.

SERGEY: 26:39 I'll play around with objects in my environment, understand more about how the world works and use that to sort of build up my body of knowledge so that when I'm later on placed in some new setting, hopefully I've learned enough from many past situations I've been in to do something reasonable in this new setting. And that would be the transfer and the transfer as in all learning systems comes from seeing a sufficient breadth of experience. So, if you have enough breadth, if you've seen enough variety, then you're ready for anything. So that's the dream.

SERGEY: 27:01 The reality right now is that this is kind of an early step in that direction. So right now, the robot learns about one particular environment. So right now, it spends a few hours playing with a door, moving it this way in that, and it can open that one door. One of the things we want to do next is actually scale this up. Maybe we have multiple robots in the lab downstairs. We have six different robots. Perhaps we can have all of them playing with different kinds of doors and maybe then we'll see that when we give it a new door, it will actually generalize to that new door because it's seen enough variety. But the system right now is not there yet.

CRAIG: 27:27 In this case. Is it important to have a system attempting the same task in different environments in order to force it to generalize?

SERGEY: 27:36 Yes, I think it is. And I think it's not anything that's actually specific to AI. I think it's just a general property of learning. So, if you were to learn to play tennis and you were to always use the same exact tennis racket every time you play tennis, and then I gave you my tennis racket, maybe you would have a little bit of a hard time with it at first. But humans generally don't find themselves in those situations. Why? Well, because humans living in the real world and the real world naturally provides variety. That example of a tennis racket is ridiculous because there isn't a tennis player in the world who has only ever played with one tennis racket because the real world forces you to deal with diversity.

SERGEY:		So yes, that variety is key to generalization and transfer. But I also think that in robotics in the long run, that won't actually be a problem because robots exist in the real world, the same real world that we exist in and that real-world forces that diversity on you. You can't escape it. Our working assumption is that if we build sufficiently general algorithms, then all we really have to do once that's done is to put them on robots that are out there in the real world doing real things and the variety of experiences will come to the robots because they're in the real world just like we are.
CRAIG:	28:31	This learning. How is it stored in the system? Is it in the weights of the perceptrons or whatever?
SERGEY:	28:37	It's a good question. I think it's actually a very natural question because we think about, well we ourselves have memories and then we also have skills. Our skills are not quite memories and our memories are not quite skills. So, what's going on? The way that it's done right now, the sort of very mundane answer is that it stored in the weights. For a reinforcement learning system, it's actually a little bit more complex because for reinforcement learning system, what you usually do is you store all of the experience, like literally all of the images, all of the actions that the robot took - basically, like get these big long movies - and you store the weights. And the reason you do that is so that when you see some more experience and you update your weights on that new experience, you don't forget everything that you have before.
CRAIG:	29:11	And where is that stored?
SERGEY:	29:12	On a big hard drive.
CRAIG:	29:14	In a database. In what form?
SERGEY:	29:16	Just literally like movies.
CRAIG:	29:17	And then there's a search function that ...
SERGEY:	29:20	So the way that you can think about this is that the system is constantly cycling through all of its old data and new data just gets appended to that same buffer and it's just a cycling through the whole thing, right? To use an overly, I guess human centric analogy, maybe you can imagine it like this, that when you sleep one hypothesis is that the you're kind of digesting what you saw and learned during the day and maybe that's what this thing is doing, too.
CRAIG:	29:52	On skew fit ...

SERGEY:	29:52	It's a somewhat statistics centric name. The idea is that you have the distribution of possible goals you might achieve and you want to modify that distribution to increase the likelihood of sampling. Interesting new goals. Modifying a distribution in this way is sometimes referred to as a skewing.
CRAIG:	30:05	You talk about self-supervised learning where the system is setting its own goals. Is that right?
SERGEY:	30:12	Yes.
CRAIG:	30:12	Can you give me a case?
SERGEY:	30:14	Yeah. Let me try to explain this a little bit. So, what I said before is that the framing phase where the robot isn't given a particular goal, it's a little bit like a play phase. But it's still directed. It's just directed by the robot. So, the robot essentially imagines something that might happen and then tries to figure out how to make that happen. And of course, imagining things that could happen requires some understanding of what are realistic situations in the world and what are not realistic situations. So, I can set a goal for myself, I can say, I'd like to make this cup levitate. That's going to be a very difficult goal for me to reach because in this universe, that's just not a realistic situation. But if I set myself a goal, I want this cup to be five centimeters to the left. That's something I can do and I can learn how to do it and I can practice and that will teach me something about the physics of this cup.
SERGEY:	30:52	So you have to have the robot set goals that are feasible but also goals that are interesting. So, if I remember that this cup can be in different places and I just keep playing with the cup forever, I'm not going to learn about all the other things in the world. The core idea behind the algorithm is that you have to use your past experience to learn what kind of situations are physically possible, but you have to sample from that distribution of possible situations so as to emphasize the more rare outcomes. So, there are many things you did in the past, you want to generate new possible outcomes that are similar to what happened in the past, but similar to those places that happen rarely.
SERGEY:	31:24	So if the robot opened the door 10 times and pushed the cup once, it should sample additional goals involving moving the cup rather than opening the door because they've already spent enough time opening the door. That's the basic idea behind the algorithm. And there's a little statistical trick that makes this possible. So, during the training phase, the robot acquires all this knowledge on its own.

SERGEY:	31:41	So the way that it figures out what might be possible is by going into the world and playing with things, which it does by initially setting some goals. So maybe the first goal is very simple because it knows nothing. Maybe it's first goal is to move its arm a little bit to the left and moves the arm a little bit to the left. There's a little bit of randomness, so maybe it moves it a little further to left that it attended. So now it thinks, oh I'll set an objective a little further to the left. So, let me set that goal because I've only seen that once it goes there, there's a little bit more randomness and maybe it goes a little further to the left and then accidentally pushes the cup. So, then it says, oh now I see that the cup can move, but that only happened once. So that's rare. Let me try that again. And it just progresses like this.
CRAIG:	32:13	That's fascinating. And obviously you've watched your robots do this. Does it happen very slowly. Like you have to come back three days later and see what it's done. Or can you watch it in real time figuring this stuff out?
SERGEY:	32:25	You can watch it in real time. You have to be rather patient. Right now, these experiments, depending on the particular scene in front of the robot, take between six and 12 hours, somewhere thereabouts. So, you're talking about being pretty patient and we do run these experiments unattended, so we have everything set up with safety boxes and so on so that we can just turn it on. Students can go home, watch it remotely through the camera and it's just going there and learning. Oftentimes we run these things overnight, so we can come back in the morning and see what it figured it out.
CRAIG:	32:49	Wow, that's fascinating. And why do you call it reinforcement learning without the reward though?
SERGEY:	32:54	So the term reinforcement learning, it's a complicated term. It originated in psychology to refer to a very particular biological phenomena where animals learn through successive reinforcement. But these days it's been abused a little bit to mean many different things. And in computer science that generally means any kind of iterative learning-based approach for control. So iterative means you try again and again, learning based means that something's being learned and for control it means that it involves making a decision. I don't think reinforcement learning is a very good term for actually describing this, but that's the term that's kind of the community has settled on.
CRAIG:	33:26	The algorithm. Is this one algorithm that is driving this in skew fit?
SERGEY:	33:30	Yes.

CRAIG:	33:31	I'm just curious how many lines of code in that algorithm.
SERGEY:	33:34	It's not that much. I mean certainly compared to designing a standard modular kind of classical robotic control system, it's much less code than that. Partly because everything is done end to end. Right. So, the images come into the algorithm, the motor commands to come out and there isn't really any other machinery that's needed except for some very low- level controllers to actually figure out the voltage on the motors and so on.
SERGEY:	33:54	So engineering wise, it's actually a very simple system. Now that's somewhat hides the conceptual complexity - modest number of lines of code, but they have to be the right lines. And in practice actually implementing these kinds of algorithms properly can be pretty difficult for kind of a, maybe a slightly ironic reason actually, which is that learning based methods, you know, they're adaptive, they get better and better with experience, which means that if there is a bug, some kind of mistake in your code things will still kind of work, they'll just work worse.
SERGEY:		So, the algorithm will sort of learn around that bug. It will not be as good as if the bug wasn't there, but it will still kind of work. And that makes diagnosing problems very difficult. If you have a bug in, like a device driver, the whole computer crashes, you know that there's something wrong. But with this thing it's produces some answer. It just takes twice as long to get there and it's maybe only half as good.
CRAIG:	34:37	In terms of scale, are you talking about 500 lines of code, 30 lines of code?
SERGEY:	34:42	Probably thousands, if I had to guess.
CRAIG:		These thousands of lines of code in this algorithm are all manually coded or is there a certain amount of AutoML that's involved?
SERGEY:		No, for all this stuff it's all written by hand. I mean a lot of it is reused, like not all the components are built bespoke for this particular project. Many of them are reused from our previous projects and there are some basic components like for example, the architecture of the neural network. We basically use the same one in all of our work, so that doesn't really change.
CRAIG:	35:17	I've been talking to a lot of people about this arms race that everyone talks about and how much do we know what's happening in China or Russia for that matter and how much do they know what's happening here. Do you have a view on that?

SERGEY:		It's a good question. I think that from my perspective, I would be very surprised if there is a major breakthrough that comes out of a lab that is not actively involved in the scientific community in this particular field.
SERGEY:	35:38	There is a tendency among scientists when they release a result to emphasize the things that are new, especially things that are radical about that result, but in reality, every result builds on prior work. In fact, it usually builds on prior work very, very closely. So, the more realistic view of a scientific result is that it's exactly the same as what someone did before with some small modification. Basically, every major result is like that from the most famous scientists in history to today and for that reason, I don't think anybody will be caught off guard, at least among the people that actually work in this area by something that's sort of is the result of decades of secret research. I think that the ideas are all in the air and while there might be small local things that people can come up with in secret on the whole, it's not like there's going to be a year's long gap somewhere. I just don't see that emerging.
CRAIG:	36:20	And do you have a view on this competition that people are starting to worry about between, particularly China and the US, but also Russia. I mean you read Russian, I would guess. Are there papers that you read in Russian that aren't published in English?
SERGEY:	36:37	No. The best papers from Russia are written in English right now. And I think that's the case for papers from China as well.
CRAIG:	36:42	That's right, yeah. So, unless there's a massively funded, which is possible in China, kind of Manhattan project to reach some goal, we know what's happening in China, at least at the basic research level.
SERGEY:	36:55	I think we do. And I think that the massively funded Manhattan project, I mean, I might sound naive in this regard because I'm certainly not an expert in political science or in economics, but I would say that a Manhattan project style effort for AI would be highly inefficient because it would be difficult to get the best people to retain the best people. And without the best people, it'd be very difficult to make substantive progress that actually pulls ahead of what the people that are working out in the open are doing.
CRAIG:	37:19	That's it for this week's podcast, and I want to thank Sergey for his time. For those of you who want to go into greater depth about robot dreams and robot touch, you can find a transcript of this show in the program notes. I've put hyperlinks in the transcript to make it easier to read further. You can also find a

link to our Eye on AI newsletters. Let us know whether you find the podcast interesting or useful and whether you have any suggestions about how we can improve.

The singularity may not be near, but AI is about to change your world, so pay attention.