| | | |
|---|---|---|
| DAISY THE GREAT: | 00:06 | MUSIC |
| CRAIG: | 00:07 | This is Craig Smith with a new podcast about artificial intelligence. Each week I speak with leading researchers in the space and this week I talk to Pieter Abbeel, one of the world's foremost experts on learning systems for robots. Pieter is a professor at the University of California, Berkeley, where he runs the Berkeley Artificial Intelligence Research lab. With colleagues, he has also founded Covariant.ai, a company that builds intelligent robots for industry. We spoke about his work in one-shot and few-shot learning, strategies for robots to learn from limited demonstrations, and about robot memories and the prospect of robots with personalities eventually assisting in the home. I hope you find the conversation as instructive as I did. |
| CRAIG: | 01:01 | To begin, I thought maybe for the benefit of listeners that don't follow the intersection of AI and robotics necessarily, and a lot of people think of AI as robotics in the popular culture, if you could sort of give a brief summary of how AI has been introduced into robotics, kind of the history and where it stands today, and then we'll talk about where it'll go. |
| PIETER: | 01:26 | Sure. Yeah, so if you think about the kind of interplay of artificial intelligence and robotics, a robot really has two parts to it. There's the physical and there's the brain of the robot and the brain of course is the AI part. The physical, and I've seen a lot of innovation, but I would say has been pretty much the same for many years now. If you go look at car manufacturers, you go look at electronics manufacturers and so forth, you see a bunch of robots doing all kinds of things, but these robots haven't really changed. They are robots designed to be really good for high precision, repeated motion, doing the same thing over and over and over and they're extremely good at it. Now when we think about intelligent robots, that's typically not what we would think about. When we think about intelligent robots, we don't think of always doing the same thing. |
| PIETER: | 02:06 | We would think of maybe a robot that could, I don't know, organize our laundry or maybe do complicated surgery or maybe could load our dishwasher or maybe, you know, assemble something it's never seen before. Just be given the CAD drawings and just say 'oh, I understand now what to do. Give me the parts and I'll just assemble it for you.' And so the latter are the kind of examples of problems that we like to think about here in my lab at Berkeley, because they really challenge what's possible with current robots and specifically challenge the AI capabilities we have and give us a very good reality check about what we can't to. It's very easy to look at things like, oh my God, isn't AlphaGo like robots because robots have to make |

| | | |
|---|---|---|
| | | decisions. AlphaGo makes decisions. But it's very, very different. Because AlphaGo is in a very constrained environment |
| PIETER: | 02:48 | It's amazing. It's a big breakthrough result, but it's still on a board. And that board has specific rules and is always the same. Whereas for a robot organizing laundry, everything's going to be different every time. A robot loading a dishwasher and things are going to be different every time. A robot assembling something it has never assembled before, has to think all the way through what that might mean. And so those are very hard problems made even harder by the fact that often the real world and simulator aren't all that well matched up. And so even if you get something working in simulation it might still not work in the real world. |
| CRAIG: | 03:19 | Is that one of the reasons why you work with physical robots rather than simply simulations? |
| PIETER: | 03:23 | So there was a couple of reasons I like working with physical robots and I also have two different hats; I'm a professor at Berkeley and I'm founder of a company called Covariant. At Covariant, we build artificial intelligence for robotic automation. And there we really focus on can we make robots do things that people before couldn't make robots do. And you go to the factory, it's all repeated motion. Well with the right AI we should be able to do much smarter things. |
| PIETER: | 03:48 | At Berkeley, it's not so much about can we get this or that working. It's more about, can we advance the state of artificial intelligence and can we use robots as a way to measure our progress. And often it's very easy, you think make a lot of progress, you know, amazing result and then you try it on a robot and it doesn't work because the real world is just very unforgiving. And, in fact, very often, you know, I just see a three year old do a bunch of stuff and say, well, why can't the robot do it? And it gives you a real reality check how far we are still away from having intelligent robots. And so the robot at Berkeley really forms kind of a challenge and a reality check and also motivates a lot of things we think about. |
| PIETER: | 04:23 | Maybe say, oh, we should just go from an image to a representation of a few coordinates of the things that are in the image. But then you say what if I'm dealing with liquids? I can’t just summarize that in a few coordinates. What if I'm dealing with laundry? That's not just a couple of coordinates. It's a lot more detailed. And so often you get challenged very quickly once you try to do interesting things with a real robot. And then the plus, the extra added bonus is, with a real robot, if you get it working, it's often useful. If you can really make something work, people can benefit from it. |

| | | |
|---|---|---|
| CRAIG: | 04:53 | You've written a lot about one-shot teaching through video or demonstration and self-play. Which of those is likely to be implemented in the real world once the hardware has reached the level that they can operate? |
| PIETER: | 05:08 | It's a very interesting question, so which one will have the most impact and if you look at the papers we write, you might be inclined to think it must be about this one or that one or yet the other one. And that's because a lot of research tries to kind of isolate specific capabilities and see how much progress can we make when we look specifically at few-shot reinforcement learning or few-shot imitation learning or maybe hierarchical reinforcement learning and so forth. |
| PIETER: | 05:33 | In practice, when we want to get something working, usually you have to bring a lot of these things together. And it's not as separate as you think from a lot of research papers where it's really focused on something very specific to try to make progress in that specific axis that we're studying. I think about the different components. I think a lot about also how the robot stack up against humans. |
| PIETER: | 05:52 | Reinforcement learning is great. Learning from your own trial and error. But it takes forever when you start from scratch. But humans don't really start from scratch. They've learned other things before. They might have learned other things implicitly through evolution and it might be in our DNA that we're ready to do something more easily, thanks to evolution. Humans watch other people and are able to imitate. Actually they're able to understand how what somebody else does maps onto their own body and how they would be doing it, which is actually a very complex type of imitation because you've got to think about more than just I'm already doing it but somebody's puppeteering me and now I'm going to do it myself, which is often how robot teaching happens, to actually watching a different person, different body doing it. And so the reason a lot of the recent work I've been doing is on few shot learning is because I think that's where there's a lot of opportunity. |
| PIETER: | 06:36 | If you think about standard imitation learning, standard reinforcement learning, it's great. I mean with the limit of infinite data, certainly it gets very far. But to make things practical, you want a robot that when given a new task can immediately learn the new thing and can build on everything it's learned before. And so the assembly set up is a good example. Once the robot has learned to assemble thousands of things, the thousand and first thing, let's say, should be much easier. It shouldn't take as long to learn that. And so that to me is a very intriguing question. How would you somehow index into that past experience. And that doesn't mean you need to keep a |

video around, but you somehow typically in a neural network somehow store things about that past experience that allow you to more easily learn something new.

CRAIG: 07:17 That's one thing I don't understand about what you're doing. Neural nets remember in the weights, right, of the layers. With something like few-shot learning or one-shot learning, or imitation or self-play, any of that, where is that experience being stored? By extension, Bert - is that the name your robot?

PIETER: 07:38 Brett

CRAIG: 07:38 Brett. If it's been doing self-play for example, is that knowledge accumulating or do you reset and start from scratch every time you start a new experiment?

PIETER: 07:48 Yeah, the question, how do we accumulate knowledge and the standard way is it's accumulated in neural networks, kind of feed forward neural networks that are often used in computer vision, let's say, for recognizing what's in an image. So you feed it a lot of examples; input is an image, output, some annotated version of the image. Maybe it says it's a cat. In this part of the image, a dog, and that part, and so forth. And the neural net gets trained - automatically make those predictions about what's in the image. And after all the data has been passed through, all of the experience in some sense is summarized in the weights, the strengths of the connections between the neurons in a neural network. It's not the only way to store information though. There are other ways to do things. For example, you might explicitly keep some things around. So sometimes in an imitation learning, we might say, if you want to do one-shot imitation learning, what it really means is that if we're given a video of a demonstration, we should be able to index into that video, be able to do the right thing now.

PIETER: 08:38 So give me one video and I can now also do it. But I get to index and refer back to that video very explicitly as I'm doing my own execution to know what to do. So for example, the first one-shot and a few-shot limitation results, it was by Rocky Duan, one of our former PhD students here and was also at OpenAI, now a founder of Covariant, was doing exactly that. It had learned to index into a single demonstration to understand what to do when it needed to solve the same task, but of course not exactly the same task, not the same motions. It's just maybe somebody, you know, shows how to set a table in a certain style and now you have to set another table in the same style and you get to index into that video to understand what you need to do. But that's one way.

| | | |
|---|---|---|
| PIETER: | 09:15 | Then in later work that, I'm still not clear which in long run is going to be the right way to do it, another student here, Chelsea Finn, showed that it's possible to also learn from one demonstration by watching that one demonstration and doing updates to the weights to the network and have the network be right to re-execute on that task after that. And both of them work great. So it's kind of interesting to see that there's these two different mechanisms and both of them seem to work. And then a whole other line of work, which is a little less active these days but was very active a couple of years ago and these things always come back at some point was the work on neural Turing machines. So this is work from Alex Graves and collaborators out of DeepMind essentially setting up a neural network that explicitly has access to a memory, and it can store things away in a memory and retrieve things from that memory to solve problems. |
| PIETER: | 10:01 | And so there's all these kinds of things happening and it's not that well understood yet what is the right way. And probably it's a combination of all of these. Certain things maybe should be stored explicitly as experiences that you can index into and re-watch the video, so to say. And other things maybe you want to internalize in the weights and it's not very clear. |
| CRAIG: | 10:23 | Just on the question of robots that are working on self-play, do they accumulate that? Are you indexing or storing that knowledge somehow? |
| PIETER: | 10:31 | So when I think about self-play, let me take a step back maybe and, kind of, what context it becomes important. So, when I think about self-play, we're really thinking about reinforcement learning. Systems that are learning from their own trial and error. And often if you just run reinforcement as is, learning is pretty slow. But if you can formulate your problem as a self-play problem, meaning your agent plays against itself, learning is often a lot faster. Why is that? If you just have to learn something without having self-play, just trying to learn, let's say learn to run or learn to stack a block or maybe learn to cook a meal, the feedback in reinforcement learning is some scoring of how well you're doing. But that's scoring doesn't come in very frequently. |
| PIETER: | 11:09 | Once you complete something you might say 'well done' and if you fail then it says 'not well done.' And so the feedback is very sparse, makes it very hard to learn. The best analogy I think a lot of people know about is training a dog. When you train a dog, if you give it at the end of the day just the feedback, 'hey dog, this was a great day' and another day you say, 'well, this is not a great day,' it's not going to learn anything. Because the day was so long, it doesn't understand what was good or bad about it. |

And that's really the challenge in reinforcement learning. The more feedback you give, the easier it is to learn. That's why you talk to your dog the whole time through to say, ‘well done, well done.’ ‘Oh No, no, no.’ And that way the dog learns more quickly. Same thing for our reinforcement learning agents.

PIETER: 11:42 The more feedback we can give, the faster they're going to learn. Now the question is how to give that feedback. Do you want to really sit there and continuously give feedback? Well, that kind of defeats the purpose. Might as well give demonstrations, which is very tedious, too, but at least you're showing how to do it right. And so if you don't want to sit there the whole time giving feedback, then probably you typically write some piece of code that gives feedback. And it's going to be very sparse. But if it's a game like Go or we've done some work where there's games where there is two robots that sumo wrestle each other, they sumo wrestle each other and what happens is initially they're all pretty bad but from wrestling each other when the same neural net plays against itself, half the time it wins, half the time it loses.

PIETER: 12:18 And that's the way you maximize signal and you learn all the time. Compare that to, you learn to sumo wrestle and you wrestle against a pro sumo wrestler and they always beat you and you're like, okay, no matter what I do, I'm always going to lose. You learn nothing. There's nothing to be learned. You're always going to lose. That's what you get. But by playing against people of your own level or playing against yourself if you can, and make some slight variations in how you play, you can see what variations help you and gradually build up. And so that had really nice results with robots, especially wrestling each other, had some nice results with robots learning to play soccer, penalty kicking and goalie.

PIETER: 12:51 Then there were some nice results DeepMind had with AlphaGo Zero rather than the original AlphaGo and then OpenAI actually before that had some new really nice results with DotA2 also learn through self-play. So I think whenever you can throw things into self-play you are essentially maximizing the signal you give to your reinforcement learning agent. It's not always clear how to do it for many problems. It's not clear. Imagine you want a cooking robot, how are you going to turn that into a self-play game. I'm not sure. If somebody can do it, they should do it and it probably will learn really quickly. But nobody's really figured out how to generally turn things into self-play.

CRAIG: 13:21 And it may be a combination, right? You may in training an AI to control a robot, you may do self-play initially and then do something – one-shot or few-shot imitation.

PIETER: 13:33 You might bring them all together. Absolutely. So, when you think about, okay, few-shot imitation or few-shot RL and self-play, you could imagine that you're trying to learn to play a wide range of games and the rules could change on you a lot or the type of player you play against can change a lot. You could do few-shot learning where you immediately adapt to the style of your opponent to more easily beat that opponent than if you had a fixed strategy that does not adapt.

CRAIG: 13:58 Yeah. And one of the things that you referenced in one of the talks I listened to was, I think it was at OpenAI, to train this hopping robot simply through verbal cues.

PIETER: 14:08 Yeah. So I think the work you're referring to was some work by Paul Christiano at OpenAI showing that when you want to teach a hopper robot to do back flips, well how are you going to do that? How are we going to specify what is a back flip? In fact, I went through the same challenge as a PhD student. We were working on autonomous helicopter flight. How to even specify what it means to do all these tricks. You're going to type out some reward function to score it? Back then we hired a professional pilot to demonstrate and we learned what it means to do a back flip, to do a front flip, to do a loop. All of those things and in this new work, what Paul was looking at was essentially instead of getting an expert, can you do it with scoring it. So being live in the loop, saying this was better than the other one, and then when it does underneath is it'll then say, oh, the reward function I'm supposed to optimize must achieve higher score on this one than the other one and over time it learns the reward function. Just much like we learned the reward function from the human pilot demonstrations at the time. This reward function was learned from feedback of comparisons between attempts.

CRAIG: 15:05 And on the hopper back flip, how was the signal given?

PIETER: 15:09 So for the back flip specific one, the hopper doing back flips, it would just do two runs and you would say which one is better.

CRAIG: 15:17 Right. Yeah, because you can envision getting a robot or an AI that's controlling a robot to the point where it would be in your house. But then if you could use verbal cues or some mechanical interface that would give the cues.

PIETER: 15:30 Absolutely. You want it to learn new things when it's in your home. You don't want it to be just a fixed machine that just does

| | | |
|---|---|---|
| | | what it's factory programmed to do. You want it to be a machine that comes in and that knows your house and your preferences are going to be different from anybody else's house, anybody else's preferences. And you want it to be able to few-shot, hopefully rather than 10,000 or a million shots, hopefully few-shot adapt to what you care about. And you exactly want to give that kind of feedback. Just what's better, what's worse is only one bit of information. It's not a whole lot of information going, but if you could talk to it, people look at that, too. You, essentially when you talk to it, you're giving it some information about how to solve the problem or about what the reward functions that you really care about and then it can start optimizing against it. |
| CRAIG: | 16:11 | Yeah. How long do you think before all of this study will be integrated in a robot that then can actually operate it in that kind of setting? |
| PIETER: | 16:20 | Yeah, I think it's a very good question and from when I started at Berkeley, this was in 2008, a lot of my motivation was there was this beautiful video out of Willow Garage, essentially PR1 robot showing that it can do a lot of the chores that we wish robots would be doing for us. But the catch was that this robot was teleoperated and artificial intelligence was not ready to make it happen. But the robot was ready to just like Brett here, which is the successor of that robot, the Berkeley Robot for the Elimination of Tedious Tasks. We named it that way because we want this robot to come and help and make sure we don't have to do all these things, cleaning, organizing, cooking, the robot can do it, laundry, all of that. The first thing we did when I got to Berkeley, was exactly that. And we said, let's try to solve laundry folding. |
| PIETER: | 17:00 | And we pushed it so hard for two years we pushed super hard in it. At the end of two years we had a demo that people were completely surprised by that this was even possible. And a robot organizing a pile of towels very reliably, succeed every time. But it was two years of work and two years of really, really good students working on it and it was still a very constrained environment, very constrained setting. It was towels and there's many other things than towels. It was using that specific table. Towels of, you know, not any size and a limited size, all those things, right? Then you're like, okay, wow, this took a long time. This is 2010 we got those results. |
| PIETER: | 17:34 | But the dream always stayed there. Like we really want to get robots into the home and adapt to every possible scenario. |
| PIETER: | 17:40 | Around that time, a little later, we started realizing there's only so far we can go if we need to do everything ahead of time |

ourselves, so we really need to put even more learning into it. And I mean of course it had a lot of learning already because otherwise you can't see what's around it. Computer vision has been doing learning for a long time. But to have it learn even more learned behaviors and that's why around to does 2011, '12 we really started pushing hard on the deep reinforcement learning direction. A robot can automatically keep improving. However it remains really challenging. So I think the best way to think of it, what makes some things easier or hard for a robot? It's not exactly what's easy or hard for people. It's more how much variation is there in the robot environment and the objects it has to deal with.

PIETER: 18:18 And the less variation, the more feasible it is. And it might be hard for humans. Like a robot might do higher precision painting but it's always painting that same car and the exact same way. So there is no variation in its jobs. And it will keep doing that thing over and over. Something really simple like you know, everybody knows about Amazon's picking challenge, right? You need to pick things out of a shelf or out of a bin and put in another bin. You're like, that seems so much easier than painting or welding. I mean clearly painting and welding are much harder in principle, but getting a robot to do painting and welding? Long solved problem. Getting a robot to pick something out of a bin, that's still a challenge. And so it's this big mismatch and that's where personally I see like, okay, I'd love to get robots in the home and I can see it happen some point in the future.

PIETER: 18:57 But the more directly solvable problems are when you still impose a bit more structure on your environment. And so that's exactly why for Covariant we're focused on more structured environments like factories, warehouses and so forth. We have a bit more control. A robot has to be smart, but it doesn't have to solve like coming into a completely new house it's never been in before with a person with completely new preferences of how things should be done and understand all of that very quickly or people will go like 'this robot is useless. It doesn't do what I ask it to do.

CRAIG: 19:30 Part of what you're expressing frustration about is transfer learning, right? That we talked about storing knowledge that robots gain through different learning protocols, but how do you transfer that learning to a new environment or a new task. Two questions: How does transfer learning work? Is it going in and relearning something through one of these indexed videos or is there some other reasoning that has to take place.

PIETER: 19:57 So transfer learning is about learning the next thing more quickly because you've learned other things before. And some of this is a lot like few-shot learning. It's very related because few-shot learning is also about after training on a lot of things now the next thing should be quicker. And so if you look at what's happening underneath in both few shot learning and transfer learning, a lot of what's happening is that these tasks are very related that you're trying to solve and when you try to solve related tasks... In fact, let me give a very simple example, but people don't really think of it as transfer learning, but everybody does it this way. Nobody does it differently. Think about ImageNet challenge. Computer vision is supposed to recognize into a thousand categories. In principle, you could train a neural network for bird, a separate neural network for horse, for car.

PIETER: 20:37 I mean the categories are more defined often that these high level things, but, basically, you could learn a separate neural net for every one of them, but that's not what it's done. What's being done is a single neural network with a thousand outputs rather than one network per category. And so what you see there, why would you do that? Why not have a specialized network for cat and a specialized network for dog. Because when you try to recognize a cat or a dog or a horse and car and house, all those things, a lot of what you need to do is shared. And so the neural network will take in an image, which is just a bunch of numbers and process that layer by layer by layer. But a lot of the same processing has to happen for all these problems, right.

PIETER: 21:12 And if you train a single network to solve all the problems, it'll actually learn the patterns more quickly because when it learns about, Oh, I need to detect where some edges in the image are maybe, or maybe I need to detect eyes and where they are. And maybe I need to detect if something is furry or not. And so all those things are shared across a lot of the tasks and it's not called transfer there, but effectively there is a transfer happening between, the tasks. It's just trained all in one go for all a thousand categories in one big network. When we do actual transfer learning, the same thing's happening. It Is just that some categories don't exist yet. You haven't heard about cat or dog. Yet you've already heard about all the other animals. You've trained your network and because it's learned about all the other animals now when you say, Oh, here are a few examples of cat and dog, well that network already has the ability to understand a lot of aspects of what an animal might look like and just needs a small number of examples to then be able to classify cat or dog or whatever the new categories are.

PIETER: 22:04 Is there an example, sort of the most advanced example of, that you could give me, of transfer learning specifically in robotics.

PIETER: 22:11 Yeah, so the most advanced example for transfer learning in robotics, I think there are a few. They're mostly in the form of few-shot learning. So the first, kind of, big result for few shot reinforcement learning was Rocky Duan's result - a different paper than the one I mentioned earlier - in few shot reinforcement he showed it's possible that you put an agent in an environment it's never been in before and it needs to find a destination. But it doesn't know where that destination is. It's just marked by a big red mark. And it's just like navigating a maze and it has no idea where to go but then it needs to somehow find a target. It doesn't have a map. And so it has to learn to run down hallways, look around corners, look for the thing is looking for, remember where it's been before.

PIETER: 22:49 So effectively internally build a map. And so it was trained on many, many environments and then was tested in a new environment. So it was able from one or two episodes in a new environment consistently go find a treasure in that new environment. So that's one example. That was very surprising, at least to me, that that should work cause that's a very long-standing problem. Simultaneous Localization And Mapping, SLAM, is a big problem in robotics. But it doesn't even consider itself with actually finding the path controlling the robot. Another one, was also Rocky Duan, one shot imitation where it learned from one demonstration what to do. Then Chelsea Finn here did something where she also showed you can do one shot imitation, but now from video rather than from more abstract information about the state. Rocky's work, do it from the coordinates of the objects Chelsea showed it's possible just do it from video. You just give one example of a video how to do it and then the robot internalizes and does it from that video.

PIETER: 23:35 So those to me are some of the most impressive examples. In Rocky's case it was a block stacking scenarios and you show a new configuration and you scramble the blocks around and it has learned to build that configuration from one example. In Chelsea's case it was putting objects into targets. So there'd be a few target locations, for example, a plate or a bowl, things like that. And then you would put maybe a peach into the red plate and then it would know, okay, now when it's my turn, wherever the red bowl is, I need to find it and put the peach into that red bowl.

CRAIG: 24:04 On this question of retention, so you have a robot and you're going to have it do something. Do you plug your AI system into that robot or are you working with a robot with a algorithm or mass collection of algorithms that you're working with each time? So presumably in the case of stacking the blocks and then picking objects, if it's the same system, it should be getting smarter as you continue to work with it.

| | | |
|---|---|---|
| PIETER: | 24:31 | Absolutely. Anytime you teach the next thing it should learn it more quickly. That said, in practice when doing research, not all research is about that question. So I think of that as a question. A specific research question is when you learn the next thing do learn more quickly. And when we're investigating that we will want it to learn things and then see if it learns more quickly. But it's not always a question we're investigating. Sometimes we investigative a question, how fast can it learn from scratch, sometimes investigate other questions, how fast can it learn from maybe some instructions? And so when doing research, it's not so much about, I have a hundred ideas of how robots can learn fast, let's put them all in one big monster of code and then just you know, do it and oh yeah, our robot learns faster than anybody else's robot cause maybe a hundred different things on it. No, it's more about, okay, what fundamentally is going to change how fast you can learn if you have learned something else before, how fundamentally can we change how experiences get internalized when a robot watches something. And so it's not really about bringing everything in one place. It's more about fundamentally advancing specific components. |
| CRAIG: | 25:24 | Yeah. Is anyone doing that though? Trying to bolt together all of these ideas and see how far a robot ... |
| PIETER: | 25:31 | I'm sure some people are trying. I mean why not? But it's often not so clear what you necessarily achieve that way. It's often easier to understand whether you're making progress or not when being more focused on specific aspects of the learning. |
| CRAIG: | 25:42 | Covariant, can you give me a use case? What kind of systems require that service? |
| PIETER: | 25:54 | Sure. Yeah. So, let me maybe take a step back on that. So when we think about what we want to do at Covariant is we really want to bring AI capabilities into physical processes, right? Could be robots, can be other machines, but essentially where physical things are involved, bring the AI into it, allowing the robot or other machines to do much more advanced things. So a year and a half now we met with about 200 companies. Essentially trying to really understand their use cases and what it is that they care about and really what is it that they want to see solved urgently and need solutions at scale versus things that are nice to have, but they don't care as much. And then the other factor we look at as does it require real AI innovation to solve this problem? Or you know, maybe it could be solved with traditional automation and they just haven't had a conversation with the traditional automators and they should just go talk to them because they can do it. And why should we use our expertise to solve from that can be solved in a way that's long understood. |

| | | |
|---|---|---|
| PIETER: | 26:44 | We can't yet share the synthesis from all that because it's part of what we consider our advantage as a company that we have had these 200 conversations and really have seen what people care about. So it's a little too soon to say exactly what they are. But at the high level, they are all use cases where going beyond the state of the art and AI for robotics that's publicly available is needed to solve these problems. |
| CRAIG: | 27:06 | Yeah. Things that cannot efficiently be hard coded. |
| PIETER: | 27:10 | Exactly. Yeah. Repeated motion would never succeed, but even things that you see at research labs would typically not be enough. You really need to take things to the next level. |
| CRAIG: | 27:20 | When is Covariant going to launch? |
| PIETER: | 27:22 | It's a good question. Yeah. Don't know yet. |
| CRAIG: | 27:25 | Yeah. This again is going towards popular notions of robots and it relates very much to these robots in elder care. The elderly in particular want human interaction. Remember Sony had a pet dog Aibo and then they took it out of service or they upgraded or something and suddenly all the old Aibos didn't work. And you had all of these elderly people that had developed emotional attachments to these dogs. Even a Roomba, you start |
| PIETER: | 27:52 | start talking to it |
| CRAIG: | 27:54 | talking to it or acting like it's alive. Can you talk a little bit about how robots, the role that they can play in filling those other human needs beyond simply mechanical exercises? |
| PIETER: | 28:06 | Absolutely. I mean obviously the mechanical will be a big help and give people a lot of independence compared to today. I think robots can have some kind of personality in some sense, right? You can design it into the robot if you want. You can make the robot extra nice or make it humorous - whatever you want. You can probably build that into the way the robot behaves. It's not personally something I spend a lot of time thinking about, but for example, on Anca Dragan here at UC Berkeley is one of the world's leading experts in human-robot interaction and she is exactly focused on those questions. Okay. What does it mean to interact with a robot - the human and robot are in the same space. How does the robot interact, understand what the humans is trying to do? How do they really together do things that are really a partnership to get something done rather than just a machine and human and have them be very distinct. |
| CRAIG: | 28:53 | Boston Dynamics, presumably they're at the forefront of the mechanics, of the hardware. Do you anticipate working with |

that kind of hardware anytime soon because it presumably would be able to do a lot more than the robots you're working with today?

| | | |
|---|---|---|
| PIETER: | 29:08 | Yeah, so absolutely. Boston Dynamics hardware is extremely advanced. I mean, they've been pushing the boundaries for many, many years. Personally, I think it'd be very interesting to work with some of the more capable robots, the humanoid robots that they build or the dog. And when I think about doing research, you think about, well, where are the real challenges? Where does that variation come from? It mostly comes from putting new objects in front of the robot. It's not so much what do you have a slightly fancier robot or not? It's more, can you keep bringing new objects over and over and over? Can the robot adapt to them? Can it maybe tie knots in a rope, can it handle laundry, can it unload the dishwasher, can it stack Lego blocks, can it assemble something, assemble yet something else, can it clean up. And so it's really the range of tasks that I see as the main driving function for making things hard more than a robot that is a little more complicated. |
| PIETER: | 29:53 | I do think ultimately if you care about the final application, but that's different from research. Research is about advancing the intelligence you build for robots. And so it doesn't really matter if your robot physically only has two fingers. Yeah, you're not going to do in hand manipulation with a two fingered robot. But it's okay because you can think about exactly the same kind of challenges as in hand manipulation by having maybe many objects that you need to do something with. It gets similar challenges that you can make progress on. And so usually the challenges are not that problem specific, but if you do want to solve very specific problems, say I really want to solve this problem, then the more advanced your robot is, the better it's going to work. So from an application point of view, of course it becomes much more important to have robot that are as advanced as possible, also physically. But from an artificial intelligence research point of view with robots, the PR2 that we've had for almost 10 years now serves its function perfectly, and there's nothing where we like, oh, we can't do it with PR2, no, we can just give it new challenges every day if we want to. |
| CRAIG: | 30:49 | Well, that's it for this week's podcast. I want to thank Peter for his generosity. For those of you who want to go into greater depth about the things we talked about today, you can find a transcript of this show in the program notes. I've put relevant links in the transcript to make it easier to explore the subject further. Our audience is growing and I'd love to hear from more of you. You can help a lot by rating or reviewing the podcast or whatever podcast platform you use. |

|  |  |  |
|---|---|---|
| CRAIG: | 31:16 | The singularity may not be near, but AI is about to change your world, so pay attention. |
| DAISY THE GREAT: | 31:23 | MUSIC |