

SDS PODCAST

EPISODE 503:

DEEP

REINFORCEMENT

LEARNING FOR

ROBOTICS



- Jon Krohn: 00:00:00 This is episode number 503 with Pieter Abbeel, professor at the University of California, Berkeley, and co-founder of Covariant.
- Jon Krohn: 00:00:12 Welcome to the SuperDataScience podcast. My name is Jon Krohn, a chief data scientist and bestselling author on Deep Learning. Each week, we bring you inspiring people and ideas to help you build a successful career in data science. Thanks for being here today and now, let's make the complex, simple.
- Jon Krohn: 00:00:42 Welcome back to the SuperDataScience podcast. I am beside myself that our guest today is Professor Pieter Abbeel. Pieter is the world's most preeminent academic researcher on AI robotics. As a professor of Electrical Engineering and Computer Science at the University of California, Berkeley, he directs the Berkeley Robot Learning Lab and co-directs the Berkeley AI Research Lab. The papers he publishes are at the absolute cutting edge of machine learning, and we'll discuss his most exciting current research in the field of deep reinforcement learning during today's episode. Pieter's work is not only academic, however. As a serial entrepreneur, he's been exceptionally successful at applying machine learning for commercial value. Gradescope, a machine learning company in the education technology space that he co-founded, was acquired in 2018. And the AI robotics firm, Covariant, which he co-founded more recently, has raised \$147 million so far, including raising \$80 million in a Series C funding round in July.
- Jon Krohn: 00:01:50 On top of Pieter's deep reinforcement learning and robotics research, in today's episode, we'll cover his productivity tips, his top learning resources and skills for becoming an expert in AI robotics, the Robot Brains podcast that he hosts, how research and development in academia is vastly different from R&D for production

industrial projects, the traits he looks for in data scientists he hires and the skills you should learn to succeed as a data scientist in the coming decades. There will be some parts of the episode that will primarily be of interest to data scientists, particularly those interested in specializing in deep reinforcement learning or robotics. However, most of the episode will appeal to anyone regardless of background, particularly if you're interested in discovering what the absolute cutting edge of AI robotics is today. How to successfully commercialize machine learning, or just how to accomplish a ton in your career. Clearly, we've packed a lot of rich content into this episode. Are you ready for it? Let's go.

- | | | |
|----------------|----------|---|
| Jon Krohn: | 00:02:58 | Pieter Abbeel, I can't believe you're here on the SuperDataScience show. This is awesome. I've been waiting for this day for months. And here you are. Pieter, where in the world are you calling in from? |
| Pieter Abbeel: | 00:03:10 | Hey, Jon. I'm in Berkeley. |
| Jon Krohn: | 00:03:14 | Nice. All right, in California, so I was supposed to have a trip out to LA recently, but we cancelled it because the day before my trip was supposed to start, they introduced a mask mandate again. So, it sounds things are locking up a bit in California. Is that affecting you around Berkeley? |
| Pieter Abbeel: | 00:03:32 | Well, I mean, everybody's affected this past year and a half, in so many ways. Definitely, Campus has new rules every few weeks. Personally, I've kind of gotten used to just sitting at home, but this room is where I've been 90% of the time or 90% of my wake time for the past year and a half and just try to get my work done. |
| Jon Krohn: | 00:03:55 | Nice. Yeah, I understand. Same thing. If viewers are watching our YouTube version, I'm sure they've gotten used to me sitting in this room, which isn't just for the |

video recording of these podcast, I'm in here, all the waking hours of my life. And people should check out the YouTube version for Pieter's background, it looks really peaceful back there.

Pieter Abbeel: 00:04:15 Thank you, Jon.

Jon Krohn: 00:04:16 Yeah, it seems like a nice office. So, I have been aware of you, Pieter, for many years, which is part of why I was so excited for this episode. So, a couple of years ago, I started running a deep learning study group, where we decided on a material to study together. So, kind of an atmospheric, an academic atmosphere, and but mostly people who are already working professionals. We did have some academics, but we'd meet in person in New York. Obviously, this is pre-pandemic. And we would decide on particular things to study together. So, we'd read textbook chapters, we would watch videos on YouTube. And in week 14 or in session 14 of this deep learning study group, we started watching Berkeley CS 294-112, which features you. And so yeah, I've been aware of you for many years. And then I had this serendipitous meeting at the Open Data Science Conference West in 2019, which was the last conference that I attended pre-pandemic, in San Mateo, near San Francisco. We happened to have lunch next to each other. And yeah, I really enjoyed that. And you've been very kind to respond to all of my emails since.

Pieter Abbeel: 00:05:36 Well, I'm glad we met. That was a fun lunch. I can't wait for in-person conferences again.

Jon Krohn: 00:05:42 Yeah, me neither, Pieter. So, you have tons of really high-caliber workstreams in your life. You have cutting edge academic research. You teach. You have co-founded a couple of startups right now. The big one that you're working on is Covariant, which has recently raised a lot of capital. You do startup advising and investing. And you

also have the Robot Brains podcast, which is an awesome program. I've listened to a number of episodes. You have some of the absolute biggest names in the data science field on your show. You've had Fei-Fei Li, recently, Yann LeCun, Andrej Karpathy. Figures that I think are so important in this field that I have them illustrated by my illustrator from my book, Deep Learning Illustrated. And yeah, so absolutely amazing. How do you manage to keep all of these plates spinning?

Pieter Abbeel: 00:06:47 Yeah, that's a good question. I sometimes ask myself the same question, how I'm going to keep it all spinning. But I think one thing to be aware of here, it's not I start them all at the same time. I think whenever you start something new, there's just, there's a big learning curve a lot of time to be spent getting up to speed. So, for example, I became professor at Berkeley in 2008. And it took a lot of time to spin up the lab, find the right research directions for my group, but at some point, things start taking a little less time, because you have certain systems in place and you found the way you want to do things. Same with teaching, the first time I teach a class, it's an insane amount of work. But in the second time, you make some changes, but you don't do as much new work as the first time. And so, I think key is there's a lot of things going on at the same time for me, but I find it manageable, because it's not like I'm trying to start completely new things, multiple things at the same time. And so, I really love my research and teaching at Berkeley.

Pieter Abbeel: 00:07:58 And for me, it naturally feeds into the other things, because when I think about Covariant what are we doing? Well, we're bringing AI into real world robotics, right? Because, traditional robots are just doing repeated motion and it's great to build the cars, but it's very limiting in what they can do. But if you make them smart, all of a sudden, it opens up many new opportunities for

robots to help out. But that's exactly what my research agenda has been at Berkeley is how to make robots smarter and so, there's a very natural flow from kind of academic breakthroughs into making it practical. And so, there's these strong connections.

Pieter Abbeel: 00:08:37 And then same with, with the podcast, Robot Brains. It's I spend so much time doing research, teaching on AI, robotics, meeting all the people that are, also working in the space, getting to know them. And it's almost like, I mean, just like us here, it's like a catch-up conversation with your friends from conferences, collaborations, and you're just excited to catch up. And so, I think there's a lot of connectivity between all these threads that is making it feasible. Same with the investing and startup advising. I really focus in on startups that are focused on AI and robotics, which I'm already spending all my time thinking about. And so, in a very short amount of time, I could give effective advice.

Jon Krohn: 00:09:24 That all makes perfect sense to me. It's kind of a high-level summary, it sounds, the trick to doing many, many things. And I agree with all these 100% and we talked a little bit about this before the show. I also do have a lot of different things going on and so, people ask me the same kind of question. And yeah, this is the thing is starting one thing at a time, making sure that you have the bandwidth for getting something off the ground, like you said. And then once you have some processes in place, you have some of the key people in place. They don't run on their own, but they don't require all of your attention. And then maybe it's time to start one other thing. And when I think about starting something new, these aren't really exactly, exactly like you're saying, they're not completely different things. They complement each other. So, I want to understand deep learning really well, because I'm doing that at my job anyway. Well, I can write a book on it and that's going to force me to know it really



well. Exactly, and yeah, so same kind of thing with the startups that you choose to advise. They can even be inspiring. There can be, it can get neural connections happening that might not otherwise have happened through these kinds of conversations. And I think the final kind of key point that you made there is, if you enjoy these things, it makes it really easy to stay involved in them.

Pieter Abbeel: 00:10:49 Absolutely. Yeah, I think that's key is really finding things you enjoyed to do for your work.

Jon Krohn: 00:10:59 This episode is brought to you by SuperDataScience. Yes, our online membership platform for transitioning into data science and the namesake of the podcast itself. In the SuperDataScience platform, we recently launched our new 99 Day Data Scientist Study Plan, a cheat sheet with week-by-week instructions to get you started as a data scientist in as few as 15 weeks. Each week, you complete tasks in four categories. The first is super data science courses to become familiar with the technical foundations of data science. The second is hands-on projects to fill up your portfolio and showcase your knowledge in your job applications. The third is a career toolkit with actions to help you stand out in your job hunting. And the fourth is additional curated resources, such as articles, books, and podcasts to expand your learning and stay up to date. To devise this curriculum, we sat down with some of the best data scientists as well as many of our most successful students, and came up with the ideal 99 Day Data Scientist Study Plan to teach you everything you need to succeed, so you can skip the planning and simply focus on learning. We believe the program can be completed in 99 days and we challenge you to do it. Are you ready? Go to superdatascience.com/challenge. Download the 99 Day Study Plan and use it with your SuperDataScience subscription to get started as a data scientist in under 100 days. And now, let's get back to this amazing episode.

- Jon Krohn: 00:12:30 Yeah, and with the work that you're doing, really at the cutting edge of academic research and applied uses of robotics. So, particularly with deep reinforcement learning to allow robot arms, for example, to be able to learn tasks very quickly, to be able to imitate human actions. Sometimes, even with only a single example of what action to take, we can call one shot learning. I can certainly see how you enjoy what you're doing across academia and industry. Are there any particular areas of research that you're excited about right now that you're taking on?
- Pieter Abbeel: 00:13:11 Yes. Very excited about a few things. But let me highlight first, maybe one that we're spending a lot of time on at Berkeley right now, which is, especially if you think about the successes we've seen in AI in the last five to 10 years, there's kind of a common pattern, which is you collect a lot of data, often labeled data, and then train a neural network to capture the pattern in the data. But mostly for this to work, this data tends to be labeled, which is quite time consuming and the things that have worked best tend to be kind of one-shot pattern recognition situations. But as you know, I mean, you're referring to the deeper out class at Berkeley, for example, that you're self-studying. I mean, in reinforcement learning, we tried to study, not just one-off decisions, but how can an AI, an agent, generate intelligent behavior where things are achieved over time, right? And so, how do we get enough data for that? And how do... I mean, we can set up simulators, but how do we get the agents to do interesting things. We can say they're robots, but how do we get them to do interesting things collect interesting data.
- Pieter Abbeel: 00:14:27 And I think that's really the most exciting question right now is how can we get these agents to collect their own data, so they'll learn from their own data collection. So, they don't just take data from us, but they are smart about what data to collect. And the interesting thing is,

even though I said they're smart about what data to collect and so forth, the direction we're going is actually what we call Play. And the idea here is if you have an agent, either in a simulated environment or a robot in a lab or real world, if they have to collect on their own, how should they even collect data? Because you don't know what they're going to want to do later, what tasks you want to give them later. You want a very general kind of intelligence to emerge.

Pieter Abbeel: 00:15:11 And the example we have for that is actually kids. If you look at kids, I mean, they just, they just play. They're just like you know? When they're really young, they just slam their toys on the floor, and bite into them, and so forth. Well, although they're playing in different ways, but kids know what it means to play. And so because of that play, later, they can learn other things more quickly. They're playing with their toys, but later, they can maybe quickly learn how to help in the kitchen, cook a meal, or maybe they can help in the garden, or maybe they can learn something in school. And so, what I'm really curious about is how we can get this notion of play into our AI agents. Let them lose in simulated or real environment and how to play. And the key thing here, that we've been working on is how to instill a notion of curiosity into those agents. Somehow, we don't want to babysit them at all times and say, "Oh, now play with this. Now, play with that." We want them to just do things on their own leverage, large scale compute to just be running and collect data that way.

Pieter Abbeel: 00:16:21 And so, we're thinking a lot about how to make our agents curious, and make them want to explore things they have not seen before. And I think that's one of the kind of most exciting recent directions where we're seeing a lot of progress actually where if you can instill them with this notion of curiosity, they can play for a while. And then you give them a task like maybe they were

playing inside a game environment. And then after some play, you say, "Now, maximize the score in the game." And now, they're for the first time exposed to the score in a game and they learn to play the game really, really quickly because they've had this experience of playing in that same environment. And so to me, that's really exciting. I mean, it's still the early days of that kind of research. But to me that will open up a lot of new opportunities if we can do this right.

- Jon Krohn: 00:17:09 That sounds really interesting. So, if we have a really simple deep reinforcement learning algorithm like a deep Q-Learning Network, the way that I teach students to code them up, or so like in Chapter 13 of my Deep Learning Illustrated book, we go through a hands-on code demo where we created a deep learning network. And the agent there, learns to play a very simple game, like the carpool game and when in order to allow to learn, we have this exploration rate. And so at the beginning of gameplay, it takes completely random actions. And through those random actions, it starts to learn some of the cause and effect relationships like, "Oh, moving right, pressing the..." I mean, it doesn't actually press a keyboard, but the action of pressing the right button that the computer can do or the agent can do, it causes the cart to move to the right and left to the left. So then, over time, we can decay this exploration rate and the agent can start to make use of the information that it's learned to say, maximize point score in the game. This curiosity that you're describing, is this, I'm guessing that this is more clever than just taking random actions. It sounds like there's probably an opportunity, instead of taking random actions to be taking actions that are as yet unexplored or might have an unusually interesting outcome.
- Pieter Abbeel: 00:18:43 Right. So, first of all, Jon, you're absolutely right. There's a very close connection between play and exploration

because both are trying to get to know the environment. And when I think about the difference, when I think about the work and exploration, I think about making the agent a little bit more curious while already keeping in mind what you actually want to do. And so, your agent is already given a task, is already supposed to do well on it, and there's exploration incentives to make sure the agent doesn't get stuck in local optima, and this can perform poorly on the task. When I think about play, it's very similar. You want the agent to not be stuck in local regions of the environment, but it's even more explicitly so that you want it to ideally explore the entire environment. And it kind of opened it in a way, such that when later you give it a task, it can effectively, of course, there's a lot of machinery in AI to do this, but effectively it can be like, "Oh, if that's what I'm getting a high score for, for hopping onto this thing or for jumping off of that thing or for maybe, gathering this kind of objects, oh, but I've done that many times before, while I was playing. I just didn't know that I was going to be rewarded for that. Now that I know that, I mean, I know exactly where to go do that."

Pieter Abbeel: 00:20:11 And so not saying it works exactly as I just described, it doesn't work with that kind of exact recall. But the way it works is actually, well, we don't know how it's going to work a year or two years or five years from now. There might be better methods, let's be clear. But what works best right now is entropy-based play. And so, the idea there is that as the agent collects data, let's say a DQN agent. The DQN agent is collecting data, and it puts things in the replay buffer. And there's no score, there's no reward yet, but we're going to infuse kind of a very generic reward that encourages play. And the reward we use is actually how different is the latest data collected from what's already in the replay buffer.

- Pieter Abbeel: 00:20:58 So, the agent encounters something that's very different what's already in the replay buffer is going to get a high reward. If it's very similar to things already in the replay buffer, it's going to get a low reward. And so, we're actually still running DQN or maybe other reinforced learning items like PPO and sack under the hood, but we're equipping it with this new reward function that encourages play by rewarding for experiencing new things. And it turns out that that's, I would say, surprisingly effective for the agent to learn from that, "Oh, this kind of behavior leads to new discoveries, new data I have not seen before." It reinforces that, starts doing more of that, and then ends up being pre trained, much like in supervised learning in my pre-train on unsupervised data, it's the same kind of idea. It's kind of unsupervised pre-training for reinforcement learning. And the neural network gets trained and is very ready to then quickly adapt to any kind of task the agent is given later.
- Jon Krohn: 00:22:02 Nice. That makes perfect sense to me. And, yeah, that idea of having the reward function that captures this entropy-based reward. Using the same word in the definition isn't a great definition. But comparing new memories with those already in the buffer to get new experiences, that sounds it will address the issue that you brought up about getting stuck in a local minimum. And then it will also encourage a diversity of experiences, thereby increasing the possible number of circumstances that could come in handy later, for some specific task. That's cool.
- Jon Krohn: 00:22:44 So, based on something that you mentioned, just before we started recording, I suspect that unsupervised learning in particular has recently taken a lot of your attention. So, you mentioned that you've more recently started teaching a unsupervised learning course. And it's interesting I hadn't, I typically think of supervised learning, unsupervised learning and reinforcement

learning as being three different kinds of problems that machine learning algorithms can be applied to solve. But it sounds like, I'm guessing based on your interest in unsupervised learning, that there is a lot of overlap. And I think we've maybe touched on a little bit here, but this idea of exploring before, the agent really knows what it's supposed to be accomplishing.

Pieter Abbeel: 00:23:38 Right. There are a lot of close connections. And that's part of, I mean, as a lot of my work up to about five years ago was more or less squarely in reinforcement learning, especially as it relates to robot learning, right? But about five years ago, I started get really intrigued by unsupervised learning, just because I mean, it's so natural that we need it, right? Unsupervised learning, it's tedious to label data, yes, five years ago, if you labeled enough data, you get a neural network to internalize pretty much any pattern. But the "Can you label enough data?" could be a question, then for harder problems. It might never be able to label enough data, but there's all this unsupervised data available.

Pieter Abbeel: 00:24:29 And so, I got really intrigued by that and for a while, spent a good amount of my time on an OFTD standard unsupervised learning paradigms working on new versions of variational auto encoders, generic adversarial networks, looking at contrastive learning methods. But then, a couple of years ago, I started to see, "Well, I think we can actually bring this into reinforcement learning, too." Same kinds of ideas, it's not going to be exactly the same because in reinforcement learning, the agent has to collect their own data. So, the question becomes "How do you bring this kind of unsupervised learning signal into an agent that's running reinforcement learning?" But what we've seen is that it's actually really powerful. There is the play angle, which we talked about. But there is also, if an agent is learning from visual inputs, which is typically what you want, especially for robotic agents,

there is a lot of machinery from unsupervised learning that you can think through how to make it work for reinforcement.

- Pieter Abbeel: 00:25:34 And we've shown actually, in some recent work that you can make reinforcement learning with vision inputs, just as efficient as reinforcement learning with direct access to the underlying state of the world if you bring contrastive learning into the mix in your reinforcement learning agent. And so of course, we haven't done this in full generality yet, but we've shown this on the kind of standard, simulated benchmarks and show some really nice results on real robots learning very fast. Thanks to combining contrastive representation learning with reinforcement learning. And that's kind of just putting them both together. But then going to play in some sense is the next thing where you really think about the agent itself, doing unsupervised data gathering for a while, beyond using unsurprised or contrastive losses on top of what it's already doing.
- Jon Krohn: 00:26:30 Nice. So, to try to see if I understood what you were saying there, a deep reinforcement learning agent has been shown by recent research, perhaps your research, to be able to have as good an understanding of the real state of the world, as it can capture with video camera sensors as long as there is this contrast of representation learning that happens ahead of or in addition to deep reinforcement learning. And I actually, I don't know a lot about contrasting representation learning. I'd love to hear a bit about that.
- Pieter Abbeel: 00:27:10 Yeah, sure. So, contrastive representation learning is the following idea. Let's say you have, well, let's say you want to build an image recognition system. Now, the canonical thing would be maybe you collect a lot of data, label it and train a neural network from image input to label output. Then you might say, "Okay, can we do better?" It

might say, "Oh, well, there are some existing datasets. Maybe I pre-train on image net," or something like that. And then for my own problem, I just collect a little bit of extra data. And often, you can get away with a lot less data that you would need if you didn't pre-train. But image net was still labeled. And so, you're still relying on somebody having labeled a bunch of images to do that pre-training.

Pieter Abbeel: 00:27:54 And so, the question in unsupervised learning in general is of course, how do we get away from that? Can we just have a ton of images, pre-train on that, and then be ready to learn a classifier very quickly later. And there's multiple reasons for that. One reason is just from a pure artificial intelligence research perspective, I mean, if we want to build the most capable AI systems, it seems they should not need all the data to be labeled. It should learn a lot from unlabeled data. But also, practically speaking, it's costly to label data, so you might not want to have to do it. So, what can we do? There's been a lot of work on this space. But the thing that I think really made unsupervised learning for image recognition breakthrough and more generally, for computer vision was contrastive learning. And the idea in contrastive learning is the following: You are going to download a lot of images, as we're doing unsupervised learning. But you don't know the labels, so how are you going to now train a neural network on these images?

Pieter Abbeel: 00:29:00 Well, here's the kind of very simple but powerful idea. You can take two images. You don't know what's in them. It might be a cup in one, a dog in another one, but you don't know and your system doesn't know and you're not going to tell it. It's just two images. Now, it's an image one, you make two variations. Maybe you turn it into grayscale and maybe you mirror and crop it for the second variation. And for image two, maybe you do the same thing or maybe you do yet something else. Maybe

you rotate it or maybe you make it darker or brighter. And so, you have a bunch of schemes to make variations but those schemes to make variations as data augmentation methods, you know they will not change the meaning of what's in the image. So, you come up with a list of things, data augmentations that don't change the semantic meaning of what's in the image. And for those both of those images, you make variations and you keep track which variation came from, which original.

Pieter Abbeel: 00:30:03 And then you say, "I'm going to have my neural network process each of those variations and turn them into an embedding vector." And I want it to be the case that these embedding vectors if they come from the same original, they're close together. If they come from a different original, they're far apart. That's the idea behind contrastive learning. You do that on enough data, it actually learns really powerful representations of, well, what's in images. They won't yet know what it is, but the neural network will be pre-trained in a way that is the best kind of pre-training that's available today if you're doing a large data set. And then you can fine tune with just a little bit of data on the classification problem you care about.

Pieter Abbeel: 00:30:47 And so yes, when you said we can now do the same thing in reinforcement learning, exactly true. Because as an agent collects data, the data that the agent collected, we can say, "Well, there's data collected now. There's data collected a few time steps later," yet a few time step later, and so forth. And actually what we do is we do contrastive learning on that day, we take data from the replay buffer, we do augmentations. And if it comes from the same original, we brand it close together. If it's some different from the originals, who push it far apart.

Pieter Abbeel: 00:31:16 Now with reinforced learning, it can actually go further, because your data is collected sequentially. And so, you

have more signal, because what you can say is, "Well, data that was collected close together, in time, should be close together and data collected far apart in time should be far apart." So, it's not just, it doesn't come from this exact same original, does it come from originals that were very close together in time. And once you do that, your contrastive learning also starts picking up for your agent on things like proximity in the space the agent is operating in. Because now things that are connected with just one or two actions from the edge, you can take it from one to the next thing, well, those things will be close together. And so, the agent becomes sometimes capable of better understanding already how the world functions, both visually and in terms of connectedness, dynamically how the world works, that it's acting in. And then on top of that, of course, we put a reward learning signal to focus it on the task or we put a play signal to make sure it collects the most interesting data possible.

- Jon Krohn: 00:32:22 Very interesting. That is exciting and I can see how valuable that would be. All right, so you've covered lots of exciting aspects of your current research. Without going into proprietary details, how do you apply your research at Covariant?
- Pieter Abbeel: 00:32:39 Yeah, it's, I mean, building a company is so different from academic research, even though obviously, we need to build in the same kind of technical foundations and do a lot of research to get where we want to be. But if I think about the things we do at Covariant, it's very driven by ultimately end customer satisfaction with what we deliver. They need a working system. If we go to warehouses where we mostly deploy our robots, if it doesn't work, it doesn't matter. And what I mean with it working, I'd say the biggest difference in terms of what we need to focus on at Covariant versus let's say, academic work, is in academic work, it's really exciting when you do something that's never been done before, like your agent maybe

complete some tasks that has never been completed before by an agent.

- Pieter Abbeel: 00:33:35 But an end customer doesn't care if a robot did something once and you have a cool video of a robot doing something once. It's like, "Oh, the robot did this once. This is so cool. This is really helpful." No, that's not helpful. Videos are not helpful for the end customer. It needs to be a robot that consistently does the right thing, essentially every single time. Right? And so, and it's fine that these things are different, right? It's fine that in academia, we focus on things. It's kind of a first. Nobody's ever had a robot, maybe, I don't know. In robotics, one of the very visible projects recent years was, for example, OpenAI's Rubik's Cube, right? Nobody's ever done that before. A robot single handedly solving a Rubik's Cube, that's really cool. But from a-
- Jon Krohn: 00:34:21 Yeah. There's really fun videos of that on YouTube.
- Pieter Abbeel: 00:34:23 Absolutely.
- Jon Krohn: 00:34:24 Because the experimenters were really messing with his hand, right? And so, coming up with all kinds of ways of poking and prodding it and putting a glove on it and it's still manages to solve the Rubik's Cube. It's really fun stuff. Anyway, I interrupted you.
- Pieter Abbeel: 00:34:38 Yeah, go watch, go watch the OpenAI Rubik's Cube solving videos. I agree. They're amazing. At the same time, when you look at it, that's a typical kind of academic breakthrough because it's not the kind of thing where this hand will 99.99% of the time correctly do this efficiently. There's a certain success rate. You can read about in the paper. I believe it was around maybe 50%. And that's a massive achievement. I love what was achieved there. But it's very different from what you need

to achieve for bringing something to a customer, who wants robots to actually help out.

- Jon Krohn: 00:35:18 We're only shipping half of our Rubik's Cube solved to the customer.
- Pieter Abbeel: 00:35:25 Yeah, you say shipping, yeah, and that's exactly what is, right? You go online, you order something, you expect it to show up next day at your door, or maybe even same day in some places. But it should do that consistently. It can't be half the time. So, what we realized at Covariant is that chasing the long tail of variability that you encounter in the real world is the hard research problem. That's where, as a company, we spend our time. How did you choose that long tail? And it's a bigger picture thing. Part of it is methodology. Part of it is, which neural net architectures can actually absorb the data that we feed these architectures because I mean, if you know that maybe it's too small, it doesn't have the right kind of architecture. Where it's set up, it cannot absorb the information in the data. But also, are you collecting the right data? Are you training on the right data? And so, it's a much more comprehensive thing. Probably the thing people are most familiar with that from just press coverage and so forth is self-driving cars, right? Self driving cars has been this thing.
- Pieter Abbeel: 00:36:31 We've had demos since, well, the '90s had demos in Germany. Ernst Dickmanns had a Mercedes car drive to highways from Germany to Denmark. Then there was the DARPA races in the desert, the Urban Challenge and Google revealed their car. I mean, if you watch those videos from those days, if you watch a one-minute video, a two-minute video, it's actually not that different from what you would be watching today. But there's a lot of progress made that you don't see them on a one- or two-minute video, because you can actually in today's self driving cars, they might go an hour, two hours, three

hours, without having a human intervention. Now, that's not enough to be driverless. You need an intervention in a couple of hours. You need somebody that are ready to intervene. Otherwise, it's problematic. But that's essentially the same thing at Covariant is it's about chasing this long tail, in our case, of robotic manipulation challenges the robot is faced with because if you look at the warehouses that we're in, the automation process that's already played out in the past years is automating essentially, the legwork. Running around in a warehouse can take a lot of time. And there's robots on wheels, robots on rails, there's conveyor belts that automate the kind of large scale motion.

Pieter Abbeel: 00:37:48 But then the pick and place operations are, I would say, largely unsolved. And that's exactly what we are solving. And so we could say maybe we're kind of starting to be solved in what we're doing. And it's all about chasing that long tail, isn't it. One warehouse can easily have a million different skews, so stock keeping units. And it can't prepare by training for those exactly those million skews, but the next week, because the next week, there's going to be new skews. And there's going to be new packaging of the same skews. And the way they're appearing in front of the robot, it's not structured like in a car manufacturing plant where everything is lined up in some exact way. No, it's always different, even for the same objects.

Pieter Abbeel: 00:38:31 It's really challenging to, every single time do a correct grasp, pick a place. And so, chasing that kind of very high performance reaching autonomy, and robotic manipulation for, in this case, warehousing operations is really what we're chasing. And so, it's a very different kind of thing. But at the same time, the foundations are very similar. Deep neural networks are being trained to do the thing they need to do. Data needs to be collected to do this right. Data that's collected needs to be augmented

in various ways. You want simulation to augment real world. I mean, there's so many factors that are similar, but it's kind of keep chasing the next nine of reliability, as opposed to saying, "Hey." At academia, we would say, "Well, we have 90%, grasping success, maybe it's time to do, I don't know, furniture assembly or something." Maybe 70 or 80 or 90% there, but for real world, no. You need to keep going and get to this multiple nines of reliability, and then you provide real value. And that's what I'm so excited about is that we are actually getting there and these robots are providing real value because we're hitting these multiple nines of reliability.

- Jon Krohn: 00:39:52 This has been a hugely valuable conversation for me, and I'm sure for a lot of our audience because from watching, I'm not a robotics expert, so I just watch the cool YouTube videos of the Rubik's Cube cells. And so, I totally had this impression before today that those robots were solving it the vast majority of the time, not 50% of the time. I would have not been surprised if it was solving the Rubik's Cube 90% of the time or 99% of the time. And so, it's very interesting to learn that there is this big gap between pick and place actions in the lab that I'm seeing these amazing videos of, and the reliability that is required in production systems, like a real world factory. So yeah, very exciting that you're making adjustments to preprocessing, to model architectures, and allowing these systems to get there 99.99% reliability. That's cool.
- Pieter Abbeel: 00:40:50 Right. And so, to that point, actually, at Covariant when we have released a video in the past, you can go on YouTube and you can actually find a video that is a one-hour long recording. We sped up the video, so you can watch a bit faster than one hour. But it's a one-hour long recording, no cuts, because that's how we can show that this actually works at the reliability levels that you need compared to what might have been lucky 30-second, one-minute snippet somebody was able to record somewhere.

And that's really been the direction we've been going as we communicate our results at Covariant. It's not about 30 seconds snippets like it would be when you do a first in an academic lab of something. It's about one-hour long video. And yeah, it just works throughout the entire hour. That's a sign of reliability. And actually, we're seeing similar things for some of the self-driving companies these days that they have also started doing that at times, where they will release videos that are one hour long, uncut, to really give you a sense of, "This is not just a little highlight. This is actually working for extended periods of time now."

- | | | |
|----------------|----------|---|
| Jon Krohn: | 00:42:05 | Nice. That is a really great perspective. And it brings the idea home and we'll try to find some of those videos and put them in the show notes... |
| Pieter Abbeel: | 00:42:12 | Yeah. It's good. |
| Jon Krohn: | 00:42:14 | ... so people can check those out. So, whether you're working on academic, robotics projects, or industrial ones, it sounds like despite the objectives being quite different, there's still a lot of R&D. We're still, we're using deep reinforcement learning, neural networks, maybe even similar kinds of hardware in both situations. So, my guess is you can answer this next question of kind of generally across both the academic and industrial parts of your life. So, what is the day like in somebody who is doing AI robotics research? So, what kinds of theory do people need to know? What kinds of approaches? What kinds of software, maybe even hardware, is used in a daily basis in a job like that as a data scientist or a machine learning engineer? |
| Pieter Abbeel: | 00:43:07 | Yeah, I mean, there's a lot to be learned to get up to speed, but at the same time, I think, it's manageable and I think that's the exciting part. It doesn't feel like there's an infinite amount of stuff to be learned. So, for example, |

when a new student reaches out to me at Berkeley and they say, "Hey, I'd to get into research in this space. What can I do?" I essentially give them a list of things to study. And these are often students who want to do academic research in this case. But I would probably give a similar list to people who want to do more practical things. At least, in the beginning, there would be a lot of overlap, because the foundations are very similar.

Pieter Abbeel: 00:43:51 And so, what I point them to, I point them to deeplearning.ai, Andrew Ng's course there, which essentially lectures through all the basics of deep learning with great examples. And gives a good foundation of what is deep learning, what are neural networks, what is machine learning, and so forth, in a kind of very, almost spoon fed way, but it's a good way to get started. But also then tell them, "Look, these things are spoon fed to you, and it will all make sense. It will click." But the way to really learn about this is as you listen to the lecture, as you do maybe an assignment in this class, you shouldn't just do the thing and then change a few things and then it works and move on. You should be questioning everything because that's how you become a researcher.

Pieter Abbeel: 00:44:47 When the instructor says, "Step 1, 2, 3," it might make sense to do those steps. But then. "Okay, do we really need Step 2? Can we do this differently? Is this the right way to do it? And not is it architecting in a certain way, these are the nonlinearities in each of the neurons. Okay, well, why are these the preferred ones? Why is the instructor saying values are maybe preferable? Well, I'm going to run an experiment with sigmoids. I'm going to run an experiment with 10H and see what happens. See if I can also get it to work. Yeah, it was harder, but I kind of got it working," and this kind of constantly questioning. And so, specifically with the students what I do if they want to get into my lab, I will tell them, "I want you to self-study these materials and I want you to send me a

list of questions every week of questions you ask yourself about, 'Well, why did they present it this way? Why is it done that way? I tried it actually differently and it also works or it doesn't work.'" And that gets them into the research spirit.

- Pieter Abbeel: 00:45:48 And then from there, I might go to a more advanced class. I had a class that you referenced earlier, Jon, the Berkley Deep Reinforcement in class. The research is going to be in reinforcement learning or the new Deep Unsupervised Learning class, and do the same thing there. Again, listen to the lectures, think it through, rederive things, code things up. I mean, most of these homeworks will have coding exercises, so it's naturally part of it, but keep questioning things. Try variations. Don't just do the thing that's prescribed to do. That's the starting point. Sure, do that first. But it's not because it makes sense, it's like when somebody gives you a mathematical proof of something very complicated, you can probably check step by step that that's correct. You'll be like, "Okay, every step is kind of simple." But actually coming up with a proof is really, really hard because you need to come up with that sequence [crosstalk 00:46:41]. Right?
- Pieter Abbeel: 00:46:42 And the same thing is true with many great lectures. Many great lectures are lectured as they should be in a way that everything is logical, everything makes sense. But then you should force yourself to step out of it if you want to get into the research mode and question every logical step. What are the alternatives? And I think that's where things get interesting. And then of course, research is kind of the same thing, but to do academic research, it's about past papers, where you start questioning why people did it this way and maybe they could have done it a different way. And you take it to the next level.
- Pieter Abbeel: 00:47:18 And at a place like Covariant, it's like "Okay, this is the common practice of building a system for reliable robotic

manipulation." But it doesn't have enough reliability if we use off-the-shelf academically available methods, "Well, let's question them. Why is this considered the way to do it? Can we come up with new ideas that can do it better?" And so, I think this kind of spirit of constantly questioning, obviously, if you go to a live lecture, I'm not recommending that you constantly interrupt the instructor like every minute with a question because that might be annoying for other people. But in your head, you should be questioning, you should be taking notes about that, trying variations, I think that's really key.

- Jon Krohn: 00:48:01 That makes a huge amount of sense to me. And it is absolutely the way that I teach deep learning as well, so where people will ask questions exactly like you pointed out. "Why should we use a ReLu neuron instead of a sigmoid neuron?" And so, I can do a little demo in class of "Okay, here is the validation accuracy we get with this simple model using one or the other. But what you need to be doing between this class and next week's class is experimenting with these in a problem that's relevant to you, from your job or your studies. So, not the endless data set, again, handwritten digit data set that we're using in class like everyone else, apply to your own problem and see what happens."
- Jon Krohn: 00:48:44 I definitely think this is the best way to learn. One thing that you mentioned there, you mentioned, Andrew Ng's deeplearning.ai curriculum. And I was initially thinking to myself, "Oh, I wonder what it is about that curriculum in particular." I mean, it is a great curriculum and one of the best known. And so, absolutely, I can recommend that one as well. But as I was thinking about that, I also then remembered as I was doing research for this episode, were you Andrew Ng's first PhD student?
- Pieter Abbeel: 00:49:11 Yes, I was.

- Jon Krohn: 00:49:12 I think that was-
- Pieter Abbeel: 00:49:12 Yes. He started as a professor at Stanford, the same moment I started my PhD. And so, we matched up and yeah, worked out great for me.
- Jon Krohn: 00:49:24 Great deal. Yeah, probably for both of you. I can't imagine. Yeah, I mean, having great PhD students like I'm sure you were at that time, must have been really wonderful for him as well. So, speaking of kind of mentoring people and looking for things in somebody that you might take on under your wing and say a PhD program, what are the kinds of traits that you look for? So, we've now talked about kind of the technical background that is critical to being able to get into AI Robotics? What kinds of traits do you look for in people that you hire or that you take on in your research lab?
- Pieter Abbeel: 00:50:02 Yeah, so I think if you want to do something really special, whether it's in a research lab or at a company, you really want to stand out and make a difference, it takes a lot of commitment. It's just the way it is. And I think the only way people can be really committed to, let's say, doing the best possible academic research or the best possible innovation R&D inside a company, is if they're just not truly passionate about. It's really what they care about because if you think of it as just this kind of a job you're doing a few hours a week, whatever, it's not the same as somebody who will, who's just like, "Okay, this is so intriguing." I mean, if we can make progress on robots actually helping out, that would mean the world to me or if we can build smarter AI agents that can play on their own and be ready to learn new things, that would just make me so happy. So, I look for a lot of passion in the people that I'm interviewing and recruiting, and try to see if they really care and are excited about these subjects.

Pieter Abbeel: 00:51:17 Another part of it is you need foundations. And obviously, it's different between a company like Covariant or other companies probably and academia. In academia, there's a curriculum and the students are learning as they go along, of course, and it's part of the process. But essentially, I'd like to see somebody, who's already learned some things, both, within the context they're in. It's all about context. Some people are in great context, have very high expectations. If they're at Berkeley, I expect them to have taken the interesting classes, done well in them. Now, if they come from somewhere else, where maybe for some reason, there is not much available at their school, then I would expect maybe someone to have compute access internet and support. That they would have self-studied some materials, maybe yours, maybe Andrew's, and so forth, that they've done self-study, and really learned a lot of things. Maybe they wouldn't have as much knowledge expertise, yet, as somebody who's been in a better context. But I want to see is this person really making the best out of their situation. Because then if they join me at Berkeley or at Covariant, I know if they're, again, likely they're again going to make the best of that new situation. And I know that's going to be a real a good situation for them to be in to make a lot happen.

Pieter Abbeel: 00:52:32 And so, I really look forward to has this person been making the best out of their situation. But also, I mean, that is also absolute minimum, they need to have a basic foundation in math. They need to understand for AI these days, Python programming mostly is the thing that matters. But those things are typically not too hard to either formally learn or learn on your own. And then the additional thing I look for, especially for people who are in context, where things are easy or where there was just a lot of great classes covered. I mean, not easy, they still need to put an effort, but they can just follow the curriculum and get going. I want to see if they had some

sign of initiative. Because somebody who just kind of follows the curriculum and just checks off all the boxes perfectly, doesn't mean that when they're faced with an open-ended problem, that they're really going to enjoy that. And so, I like to see strong kind of performance on the standard curriculum, but also did they have side projects. How did they learn things on their own, build things on their own that are not defined by somebody else, but they just took the initiative to try out some things on their own.

Jon Krohn: 00:53:42 Yeah. I couldn't agree more with the points you've made. I wish this was something that, I haven't come up with a way of getting this out of an interview exactly yet. But I know early on with a new hire, if they come in on Monday morning, and they had something that over the weekend, they were like, "Oh man, I was in the shower. And I had this idea. And I just couldn't resist trying it out. I was up late last night creating this simple version of this model. And look, there's already some signal here. We are getting some results." And I'm like, "Amazing. You want to spend your week working on this now?" And they're like, "Yeah, of course." And so yeah, I couldn't agree with you more. There's something really special about people. I think we're very fortunate in a field like data science, machine learning, we have such interesting problems to solve and so, people who naturally gravitate towards these problems have this curiosity. I think there's an opportunity for these people to have such exciting careers for decades. Anyway, kind of going off on a tangent.

Pieter Abbeel: 00:54:53 Well, but I think that's a really good point. I think not every field lends itself to necessarily being equally passionate about it per se. I mean, some things, it makes sense to say, "Hey, my job is X. I do my job and I can do more of it." But that's just doing more of the same and doing more of the same is, I mean, that's often not as exciting as what we see in AI, data science and so forth

these days. When we spend more time, we don't do more of the same, we actually build out our expertise. We become more capable. We become capable of doing things ourselves as well as a community we could never do before. And I think that's a very different field. This kind of continual growth of personal and community capabilities compared to just some things that need to be done.

- Jon Krohn: 00:55:52 Totally. That was much better said than I was able to articulate. Thank you, Pieter. Yeah, that's exactly it. And it's really cool how, for whatever reason, I guess, because so much of the advances that we've had in data science over the recent decades have come out of academia, there is this sharing, there is this community. So, archive papers, conference presentations, GitHub repos, StackOverflow, we're getting in real time information that is at the cutting edge. And people come up with this amazing thing. They get a robot arm to do something that's never been done before. They publish a YouTube video right away. And people can dig into how they implemented this instantly. It's a really exciting time to be in this space.
- Jon Krohn: 00:56:39 Anyway, so kind of thinking about the way data science skills have evolved recently and how we now have this ability to access the cutting edge of our field. Pieter, what skills do you think will be valuable 5, 10 years from now? What are the things that maybe even are difficult skills to attain that are worthwhile getting started on learning about today?
- Pieter Abbeel: 00:57:08 Yeah, that's a hard question. Just predict the future. Yeah, what could go wrong? Fast moving field, no problem. So, it's a very fast moving field. And so I mean, the kind of obvious answer is just kill to be able to absorb new innovation quickly, right? When new ideas come out, get proven, the ability to look at that, understand both

what it can do now and what it might mean for things that might be possible in the near future. Maybe it's shown in one domain, but being able to think through, well, without working other domains, too, work on images. Would the same idea carry over to text? Would it work from natural images to medical images? What are the differences? Would it work on a more database structured data, and so forth. I think that the skill to quickly understand kind of the potential of all the breakthroughs that are happening, I think that's going to be there for a while to stay. And that really means having a pretty broad foundation in everything that's already there. And having seen that happen before can help a lot.

Pieter Abbeel: 00:58:20 One thing that I'm kind of excited about for the near future goes almost in the opposite direction, which is that I think we have this continuous innovation and more and more new things are happening, it's really exciting. And that's exactly the direction we just talked about. But I think at the same time, all the innovation that's already happened, still has a lot of applications that can be built around it. And so, the existing tools that we have, let's say for deep learning, and building applications, they are becoming, I would say more and more usable, even if you don't have a deep mathematical understanding of everything that's going on under the hood because this start becoming usable in a bit more of a blackbox way.

Pieter Abbeel: 00:59:14 Naturally, if you use them in a blackbox way, you're not going to invent the next generation of it, most likely, but there's a lot of applications to be built in a more blackbox way. And I think we'll see a lot of that in the next few years. Where I think what will be important is to kind of understand just like with more traditional startups and projects is understand the product side of things. Well, if we can do pattern recognition on a certain type of data, what kind of product can we build from that. And I think that is going to be really interesting. And a skill that's

going to become very important next to, of course, being able to pioneer the next, next, next generations of everything is this kind of time product building to what AI can do today in a way that is relatively easy to play around with compared to a few years ago, and will keep becoming easier. And so one of the skills I'd like to highlight always, especially to kind of more business oriented, folks is when thinking about bringing in some kind of technology, traditionally, technologies have been fairly deterministic.

- Pieter Abbeel: 01:00:24 You bring in, I don't know, a new piece of equipment, a new camera, and say one megapixel camera. And now, you have a one megapixel camera, you know that that's the quality of the streaming it provides and so forth. If you choose a product based on very clear specs and I think that's, of course, different for AI. If somebody says we're providing image recognition, but what does that even mean? They say they have, I don't know, 95% or something. Well, if they're honest about it, then what that actually means is that they have 95% on if you feed data in of the same type they have been feeding data in, that's what it would mean. And well, is your data the same?
- Pieter Abbeel: 01:01:10 And so, I think just kind of understanding that all of these AI systems are driven by data and their performance is calibrated on a certain data distribution. And as you think about building your applications, choosing the right provider, and quickly kind of troubleshooting whether your data matches the existing system, and you don't have to do any training at all, or it doesn't match, you actually need to train your own network. I think that understanding combined with product understanding what consumers, what B2B, what companies want, is going to be just as big, if not, if not bigger, I think in the next few years than pushing the frontier of the next generation of AI technology. Both will

be really big, by the way. It's hard to predict which one will be bigger, but I think both will be so exciting.

Jon Krohn: 01:01:57 Yeah, those were... you had a series of really valuable nuggets in there, Pieter. So, there were things like it sounds learning how to learn, practicing that skill is going to be hugely valuable and it's something that we can really practice that. Some people do have, I guess, some natural ability to remember facts better than others. But you can study these things, and becoming better, and I've talked about this in previous episodes, I'm not going to dig into it right now. But there are, you can look up how to be better at that and it will serve you well, in any field, including in one that moved so quickly. You touched on this idea of getting creative, being able to see cross and disciplinary applications of innovation, which requires kind of a broad understanding of disparate fields. So, that's also a great huge tip. And then at the end there, talking about just bringing the cutting edge to applications. And so you talked about this a lot earlier, with your own work at Covariant. This idea of taking this state-of-the-art pick and place capability that a robot arm has, but has low reliability, and then figuring out how to production that as that in a way that has lots of nines in its accuracy. So that is, yeah, really brilliant guidance, Pieter. Thank you.

Jon Krohn: 01:03:27 So, when you think about your own career, we're experiencing such amazing changes right now. And we have been for decades, and it's going to continue. So, ever cheaper data storage, ever cheaper compute ever more abundant sensors everywhere, interconnectivity, data modeling, innovations, and our ability to share all this over the internet instantaneously with each other. Is there anything, do you ever think about what you might want to look back on near the end of your career in terms of what you can accomplish given all of these wins that we have at our back?

Pieter Abbeel: 01:04:15 Yeah, so I mean, part of how I think about that is, I mean, each of us have our own kind of strengths. Some are just talents were born with, but a lot of it is things we've built up over time, right? And so, when I think about what do I want to be able to look back at and be happy about what I spend time on? I tend to think a lot about what are some things that I'm, relatively uniquely qualified to do, because if I spend time on those, I can have much more impact than if I, you know? Well, I mean, I love basketball and tennis. I could have a lot of fun every day doing nothing but playing basketball and tennis. And I'd probably have a lot of joy on a day-to-day basis, but when I look back at it, many years from now, I don't think I would look back at it with the same satisfaction as what I'm actually doing, which is playing to things I'm actually more uniquely qualified to do them than those two sports.

Pieter Abbeel: 01:05:13 And so, to me, I feel like the way my career has shaped up has positioned me really well to make a big impact at the junction of AI and robotics. And that's really what I look forward to, kind of push much further than we are today, both in academia, making academic breakthroughs on that front as well as bring it into the real world. And I really want to do both because I think without the connection to the real world, it's hard, I think to be fully satisfied about. Is it really progress? It's a little harder to be sure. You're really making progress if it never makes its way into real world impact. And so to me, it's hard to put kind of a specific achievement on it to say, "Okay, I want to achieve X, Y, or Z." It's more, I kind of want to push myself as hard as possible and see where we get. And hopefully, we can make a lot of progress both academically and putting AI Robots into the real world. And I think we will. I think we are already doing it and we'll expand quite quickly in the future. But it's very hard to predict how far we can get in what timeframe. But as long as I give it my best, I feel like I'll be happy with whatever I made happen.

- Jon Krohn: 01:06:42 Yeah, that was a great answer. All right. So, Pieter, being mindful of your time, I unfortunately can't ask you every possible question. We've had a huge amount of interest. I asked on LinkedIn and Twitter last week if anybody had questions for you, and boy, did they ever. So, we had 15,000 views of the post, and dozens of questions. But I'm going to go based on the ones that kind of had the most reactions. So here's one from Serg Masis, who's a data scientist. He specializes in Explainable AI. He asks, "Just as natural intelligence has many signals and states for cognition and emotion, do you think it would make sense to mimic these kinds of emotions in AGI?"
- Pieter Abbeel: 01:07:28 Yeah, I think it's a really good question. I think it kind of goes back to kind of the high level question, right? If we want to build AGI, we have one proof of concept, which is evolution on Earth has provided it in humans and reextend them and other animals. So, I think the question becomes, how many shortcuts do you want to take? And what are the right shortcuts to take? Obviously, we don't want to run, at least I don't think we want to simulate the past 5 billion years of what happened to get to an edge on in a computer. And so, in research, I think that's always the recurring question, things like emotion. Do we want to architect it in or do we want to maybe put our agents in environments where it naturally by having emotion, they will perform better. Now, of course, then they'll have to acquire it, then it might take more time. Often acquiring it, though, is more robust than try to hard code things.
- Pieter Abbeel: 01:08:32 For specifically, things like emotion, my hunch would be that we would probably acquire through learning without a very dedicated architecture tweaks. But it might have some dedicated data collection to get the kind of coverage that you need there or if it's an RL type agent, it might need some environmental situations where things like emotion really can play a positive role in its performance.

- Jon Krohn: 01:09:02 Nice, great answer. There you go, Serg. And one last one here, from a senior robotic researcher, whose name I'm sure to mispronounce. But Hsieh-Yu Li is curious about how you and your team find practical problems or real world applications for deep reinforcement learning. So, do you tend to work on a new reinforcement learning algorithm first, and then try to find practical applications or is it more the other way around where you notice that there's a problem and you find ways of devising a model for that problem?
- Pieter Abbeel: 01:09:43 Yeah, and both approaches are great. I think it's kind of like the two research approach of pushing the technology frontier further versus looking at a problem that's unsolved and see if you can get closer to solving it. And I'd like to do a mix of both. I think it's nice to have that both going on at the same time. In general, though, I mean, if we look back at the work we've done, I would say the kind of unsolved problem we're looking at is robotics. Robotics is clearly unsolved. We cannot have a home robot doing things for us the way a lot of us dream robots could be doing for us, if they were just smarter. And so there's that clear kind of unsolved problem pull, then of course, you break down into simpler problems that are still unsolved. And to me, that's often the inspiration. Whenever we feel we made a lot of progress, well, actually, is it enough to have a home robot that does the things we would want it to do? And we're like, "No. It's not." And then we're back to the drawing board.
- Pieter Abbeel: 01:10:43 And then we might try to tease apart, "What are really the missing pieces? Why can't we have a home robot doing everything for us? Well, RL is not good enough in this way, that way, that way. Okay, now we have specific things to think about that we should make progress on." Now, often, some of these things are recurring, like it might sample complexity is just too bad. And at that point, you're kind of pushing on the same thing over and

over without necessarily continuously revisiting the home robot problem. You're just saying, "Okay, RL needs better sample complexity, we know that and we can keep pushing on that in its own right without revisiting, the open problems that we want to solve long-term every time.

- Jon Krohn: 01:11:22 Nice. So, there you go. Nice clear answer. And I wasn't surprised by the answer. I had a feeling it might be a little bit of both ways. All right, so let's start to wrap things up here. Pieter, I always ask guests for a book recommendation, do you have one for us?
- Pieter Abbeel: 01:11:40 Sure. Yeah. I mean, I've learned a lot of things from books. Maybe the one I'll recommend is one that's maybe a bit outside of the technical field, because you probably get a lot of technical recommendations already. So, let me tell the backstory about this. So, I'm a PhD student at Stanford at that time, many years ago. And I write my papers and hand them to Andrew Ng, my advisor. And he tells me, "Oh, it looks great. Just a couple of suggestions." And then I look at it and the eight-page paper is more red than it's black. That's what Andrew would kindly call, "Just a couple of suggestions." And I worked through it, and I look at all the suggestions. And I'm just like, "Wow, these suggestions are amazing, I should incorporate them." And I incorporate them and the paper is better.
- Pieter Abbeel: 01:12:34 And then this process repeats again and again, but the problem was, I wasn't really seeing the pattern. So, I'm there every time incorporating new suggestions, but they are, and they're great, but I cannot do it myself, yet. It's like, and again, it's Andrew is the discriminator to some extent, and I'm the generator, and I need that continuous signal. And so, I'm like I'm not a good generator yet. So, I decided to walk to a Stanford Bookstore at the time, and I have browsed through all the books for the writing classes, and I've browsed about a dozen books. I buy

three of them, read them more thoroughly. And the one that I think helped me the most was a book by Williams, called Lessons in Clarity and Grace, and that book really helped me in my writing. And still today, I think, I mean, especially now the more I'm in some sense, in a kind of advisor/managerial role, rather than day-to-day execution on specific things. Communication is even more important and a lot of is in writing. And so, yeah, that book has had a great positive influence on me. So, Williams, Lessons in Clarity and Grace.

- | | | |
|----------------|----------|--|
| Jon Krohn: | 01:13:42 | Nice. |
| Pieter Abbeel: | 01:13:42 | By the way, yeah... |
| Jon Krohn: | 01:13:43 | That is a great recommendation. I'm sure listeners will love it. Yeah? |
| Pieter Abbeel: | 01:13:46 | I was going to say, by the way, after I read the book, I actually could understand all the suggestions Andrew made. I was like, "Oh, this suggestion here maps to this lesson and this suggestion here maps to that lesson in the book. And it was like, "Okay, it all makes sense now." Now, I can just do it. |
| Jon Krohn: | 01:14:02 | Nice. Yeah. And I mean, I obviously have no idea what your speaking style was like all those years ago, but you are an amazing communicator. You've been an incredible guest on the show. I can't wait to get this episode out there for everyone to enjoy. Pieter, obviously, listeners know that they should check out the Robot Brains podcast. But how else should people follow you? Twitter, LinkedIn or what? |
| Pieter Abbeel: | 01:14:28 | Both Twitter and LinkedIn. I tend to post updates on things I'm doing. So yeah, please feel free to follow me there. And if you have any questions, suggestions, yeah, feel free to reach out. |

- Jon Krohn: 01:14:44 Nice. All right. Thank you so much, Pieter. Have a wonderful rest of your day and maybe someday we'll have the great pleasure of having you on the show again.
- Pieter Abbeel: 01:14:52 Yeah, and maybe even have a lunch in-person again.
- Jon Krohn: 01:14:55 That sounds incredibly. Absolutely.
- Jon Krohn: 01:15:04 Given that he's a world leader in AI robotics, I am not at all surprised that Pieter packed today's episode so full of valuable and fascinating content. In today's episode, we discussed how the secrets to successfully leading myriad of workstreams are to start one big project at a time, have synergies between the projects and of course, do what you enjoy. We talked about how play and curiosity are critical to allowing deep reinforcement learning agents such as those deployed in robotics applications to be prepared for a wide variety of industrial tasks. We talked about how contrastive representation learning enables deep reinforcement learning algorithms to make the most of swathes of unlabeled data and develop an accurate representation of the environment it takes actions in. We covered how academic robotics research is focused on single jaw-dropping breakthroughs, which may occur at a high error rate while industrial robotics research is focused on developing algorithms with very low error rates that can run for hours without a mistake.
- Jon Krohn: 01:16:10 Pieter provided us with freely available courses, namely deep learning.ai Berkeley CS 285 on deep reinforcement learning and his own deep unsupervised learning lectures that can together enable you to become an expert in AI robotics, so long as you question and experiment constantly while following the courses and provided the you have strong foundations in Math and Python programming.



- Jon Krohn: 01:16:37 As always, you can get all the show notes, including the transcript for this episode, the video recording, any materials mentioned on the show, including the free courses I just mentioned, the URLs for Pieter's Robot Brains podcast, and his social media profiles as well as my own social media profiles at superdatascience.com/503. That's superdatascience.com/503. If you enjoyed this episode, I'd of course greatly appreciate it if you left a review on your favorite podcasting app or on the SuperDataScience YouTube channel where we have a video version of this episode.
- Jon Krohn: 01:17:13 If you'd a free way to shore up your machine learning relevant math skills, as Pieter suggested, one way to do that is through my mathematical foundations of machine learning curriculum, which is available for free on my Jon Krohn YouTube channel. I also have a usually pretty darn cheap Udemy version of my Math for ML curriculum, if you'd to support my work, and in return, you'll get detailed walkthroughs of the solution to every exercise. That's it for today's episode. Thanks to Ivana, Jaime, Mario and JP on the SuperDataScience team for managing and producing a such an incredible episode for us today. Keep on rocking it out there folks and I'm looking forward to enjoying another round of the SuperDataScience podcast with you very soon.