EPISODE 1193

[INTRODUCTION]

[00:00:00] JM: Reinforcement learning is a paradigm in machine learning that uses incentives- or “reinforcement”- to drive learning. The learner is conceptualized as an intelligent agent working within a system of rewards and penalties in order to solve a novel problem. The agent is designed to maximize rewards while pursuing a solution by trial-and-error.

Programming a system to respond to the complex and unpredictable “real world” is one of the principal challenges in robotics engineering. One field which is finding new applications for reinforcement learning is the study of MEMS devices- robots or other electronic devices built at the micrometer scale. The use of reinforcement learning in microscopic devices poses a challenging engineering problem, due to constraints with power usage and computational power.

Nathan Lambert is a PhD student at Berkeley who works with the Berkeley Autonomous Microsystems Lab. He has also worked at Facebook AI Research and Tesla. He joins the show today to talk about the application of reinforcement learning to robotics and how deep learning is changing the MEMS device landscape.

[INTERVIEW]

[00:00:57] JM: Nathan, welcome to the show.

[00:00:58] NL: Hi, Jeff. Thanks for having me.

[00:01:00] JM: You do research relating to the intersection of reinforcement learning and robotics. Why are those two topics interlinked?

[00:01:07] NL: So they're interlinked kind of in a vision of what researchers and society has for robots and that they're going to be able to encounter new tasks and learn them on their own. And I think a lot of researchers kind of have that as like a long-term vision for their work, and robotics has grounded in a lot of different science fiction areas, which I think we'll get into. So that's my short answer.

[00:01:33] JM: Give me the long answer.

[00:01:36] NL: Let's see. Starting in reinforcement learning, reinforcement learning is encountered in so many different inside frameworks and it's grounded in our biology and like how humans work and we want our creations to be able to solve problems like we do. And robots are somewhat naive when we just give them a physical body. And what we want to do is kind of give these physical creations, the awesome powers that we have to solve new tasks that haven't been seen before. And it comes to this dream of I can have something that'll solve all the problems that I have in my day. It can be unloading the dishwasher. It can be organizing my meetings. It can be optimizing my diet and kind of having something that's there to help you with all aspects of your life both physical and on the digital side. And the definition of robot can be kind of vague in that manner.

[00:02:32] JM: Can you give me an example of one of the researches, the research topics or research applications that have been explored in your work.

[00:02:42] NL: Yeah. So my PhD, I have been working on flying robots mostly, which is one application space. I kind of wandered into it. But the idea was, specifically, can we learn to fly a new robot or something that's very well studied like a quad rotor and can we learn to fly just by applying different motor voltages, which is how fast the rotors spin on a quad rotor? Or the other work robot I work on has a fancy ion thrust mechanism, which has some cool physics behind it. And the idea is if this thing is sitting on the ground, what do we need to do to have it teach itself to fly? And the applications for this is kind of if we have a robot that can teach itself to fly, maybe it can teach itself to pick up a package and safely deliver a package. Because my vision for this is kind of in that when you have rockets carrying things to space, there's a very common problem where if your payload is shaking inside of the rocket, that'll turn off your mass balance and can have very catastrophic effects on rocket launches. But in the space of delivery drones, like we want to make sure that we're not having crashing delivery drones especially if we scale them up. So we want to be able to have a robot pick up a new payload. Learn what is different about its dynamics and be able to do simple locomotion primitives.

[00:04:09] JM: What's the necessary setup for such a complex research application? I mean do you need drones? Do you need a lot of computation power? Do you need a bunch of students to do some kind of testing around this? What's needed for an actual setup of this kind?

[00:04:29] NL: Yes. So I think in grad school in kind of small labs, you touched on all the important ones. You touched on people. You touched on hardware. You touched on compute. And kind of a research vision I have, I would say I'm early on this quest, is you want to remove the need for all of those things.

So my first setup that I did, there's this paper that's a low level control of a quadcopter with reinforcement learning and we kind of used all these. We have a big computer and there's some C code infrastructure that is taking the reinforcement learning and sending it via radio to the quadcopter and then the quadcopter executes an action and sends information back. But the computer has a very fancy NVIDIA GPU in it that was donated to us and then there's also the important thing of we were breaking a lot of drones and doing this. You have to have a lot of hardware. And then there was me there that kind of supervised this. And I say supervised because there's a whole field of machine learning, which is self-supervised learning, which is like the idea that these systems can do these themselves. But the funny thing is like when learning to fly, these robots kind of discover different stability modes. So I could go into a lot of details on kind of quadcopter dynamics and what went into play there. But one of the things that it was doing was I realized that applying a lot of thrust, it makes it – So it's probably not going to crash, because if you apply a lot of thrust on a quadcopter, it's going to go straight up really fast. And we were in like a 20-foot room. So my supervision to prevent more damage was it would fly straight up. It would detect the ceiling and then it would come tumbling down and I would have to crash it. And that's not going to work without me there catching this robot.

And a lot of the experiments I talk about to colleagues with Berkeley at is like you have a robot and then you set up another robot to catch the dropped object that your robot does and put it back into place so it can do its next trial of reinforcement learning. So like kind of making these things more computationally accessible. So I don't need a GPU to do it. Maybe we can put this on board with the quadcopter. NVIDIA's been doing a lot of work on their, like Jetson Line and Jetson Nano to make these chips really small, which is what's happening in like the Skydio drones. They have really small GPUs on board to do machine learning. So I don't know. If you have – Maybe we can go into the like self-supervised learning side of things too if you want me to.

[00:06:50] JM: Yes. Let's go to the computation side of things, because it's easy to imagine the physical back and forth of a reinforcement learning loop. But tell me more about the computational side and getting the feedback mechanism digitized.

[00:07:06] NL: Yeah. So this is probably some of the software that I've written in my PhD that I'm proud of. So let me start at a high level of this system, and I think this generalizes to a lot of real time learning in the loop robotics, which is we have this crazy fly, which is an off-the-shelf research platform. It's very popular with researchers. It has some firmware that's open source and it has a whole radio set up. So what would be done is it would read its state. So it would be like attitude data. So something like yaw, pitch and roll, which is something you hear a lot for aerial vehicles, and then accelerometers providing a bit more information on the orientation. And then we compressed that and sent it in a radio packet and then it came back to a computer that was running the robot operating system, or ROS, and we kind of had it set up so that ROS was constantly looking for packets at a very high frequency. And then when it got a new state data packet, it would then take this code and it would interface with a trigger to a Python loop. And we had PyTorch and CUDA optimized to very quickly run the data through our like optimizer, which is the kind of the machine learning part of the problem. And I suspect we'll talk about RL, reinforcement learning, a little bit more. So that's like the very specific thing we did. And then it would compute the next action for the quadcopter, which in our case was actually a voltage. So it's a voltage for each of the motors and it would then do the inverse. So it would go in the Python loop and then to the ROS computer again, which would then send it via radio back to the quadcopter.

[00:08:43] JM: And tell me more about the stack of reinforcement learning tools that were sitting on the machines that were doing the computation.

[00:08:54] NL: Yeah. So this was pretty like low-level PyTorch. There's kind of in this loop that we were doing an iterative process. There's like a training process and then there's this runtime process. And I'm focusing on runtime now. So what we would have is I would manually load on the PyTorch model file. So it holds all the weights and biases. And before running an experiment, we would load these into the computer and into the GPU. And so what is done is essentially the state data from the quadcopter is read from the python loop. It's just read continuously. And then what happens is that state data alone gets passed to the GPU and then the GPU does a series of parallel computations where it just does a lot of – Essentially it's trying out a bunch of actions at once to see which action would be the best and then it selects the best action and sends it back.

And some of the important software things are these kind of systems can get very CPU bottlenecked quickly. So, originally, we're trying to not define that line between CPU and GPU as clearly. And what happens is your control frequency will be heavily dropped by the CPU moving data around. So we had to only send the state to the GPU and only send an action back. Otherwise that would be the limiting factor of our control frequency, which it goes to say how impressive the GPUs are at doing a lot of parallel computations very fast, because we were stimulating about 5,000 action candidates across a time horizon of 10 or 11 steps. So each candidate in each horizon step is a forward pass through a neural network. So we're doing like 50,000 passes through a neural network, a frequency of up to 100 hertz.

[00:10:46] JM: And what did you learn over the course of this research project?

[00:10:52] NL: In regards to broadly robotics and RL or like specifically what was the like –

[00:10:58] JM: Both. Both broadly and how specific – And the specific application.

[00:11:03] NL: So I'm going to change the order and start with specific and then go into what that means. So the specific contribution of this work was showing that you can do something called low level control with model based reinforcement learning techniques. And essentially the crux of why this is interesting at the very specific end, the pointy end, is the fact that quadcopter control is passively unstable. So if you have a quadcopter in the sky, if you lag or you send one bad action, that like time where it's not updating in the correct way is very likely to cause it to crash. These things could flip over very fast. The internal controllers run at up to 500 hertz. So our controller of running – We did different experiments at 25 to 100 hertz. It was much slower. So we have to be pretty precise in our actions. But it was showing that you can use a pretty computationally intensive and purely data-driven approach to fly a quadcopter at all, was the first time that anyone had done this.

And I think this is – And kind of broadening out, this is related to the idea of like locomotion and robotics. So very broadly is like if I have a new physical robot, can it learn to move through the world and interact with the world? It could be anything from like a quadruped robot. You could think like Spot from Boston Dynamics. What if they didn't have engineers to design all the precise controllers? It's like could it learn to walk in similar ways? That's what this technique could be used for.

And then also something as creative as one of my favorite like workshop papers at NeurIPS a couple years ago, this is a machine learning conference, is they some people tied motor – Connected motors and sticks together and then generated locomotion primitives just like how the stick could move around a room and like what obstacles it could get over. It's like kind of the idea of giving a physical platform the ability to move in the world in a very general way where we don't have to make any assumptions on how it can move? What the environment is? It's just creating locomotion primitives.

[00:13:14] JM: When you spoke to robotics experts in the industry, like did they have – What did they say back to you when you told them your research findings?

[00:13:26] NL: This is something I'm trying to work on and get to know more of. In writing about robotics as an industry and trying to talk to more people in industry, there's kind of like a practical versus research gap. And something I'm thinking about is like a lot of the methods that I'm researching and I'm more excited about kind of fall into the realm of science fiction. And then whether at talks or conferences, it's a lot of the like the robots are not robust or safe enough to use for different techniques.

So something I remember writing about delivery drones, and the FAA cleared delivery drones for most like residential suburban airspaces in like 2012. And the problem is, is that they're not reliable enough yet. So a lot of the research I see seems to be like a couple years ahead of what actually needs to be solved to implement these things. And I think that's not necessarily best defined as a problem. It's kind of like a mindset for research. But there's plenty of failed robotic startups because the robots like can't do the task cheap enough and with enough reliability. Obviously there's amazing use cases for where robots are becoming successful, which is like the pick and place problem is a classic one. I have colleagues at Berkeley starting a new company there and they're doing amazing things. But the broad like robot that can do any task is not good enough to be monetized. And I would say my research is trying to go towards robots, like methods that can be used for a whole bunch of different tasks.

[00:14:57] JM: Why is that? Why are you focused on multi-purpose robots? Why not just have the Unix philosophy applied to robots?

[00:15:07] NL: I would say it's because I'm more excited about it and it's easier to sell as an academic as your method is something that you want people to use. I want people to use model based reinforcement learning and I see a lot of benefits in it and I think we'll get there. But the academic mindset is very much the selling of ideas and making it as accessible as possible. But I think that that makes it so the actual translation, like the translation is used in multiple fields it's like academic to reality. It's like the translation is not really there yet for robots. And it kind of keeps me grounded in some ways. And as I plan on graduating in a couple years, that's something that I'm going to have to think strongly about, is like how do you realign the problems to make them more practical? Because impacting lives is what I would like to do. And I know a lot of colleagues were. But it's an incentive thing.

[00:16:02] JM: Did your work in the locomotion, the robotics locomotion research that you just discussed, did that provide an impetus for any other investigations that you've launched?

[00:16:15] NL: So there is a longer arc to that and it's kind of there're work that's ongoing that's kind of slowed by the whole pandemic and things like that, which is my advisor is trying to make a whole new type of robot. He's trying to make micro robots that are silicon fabricated and things like this. And the goal is – The methods that I'm investigating can be used to synthesize controllers for these robots. And we we're going to have a lot of robots and they're going to have fabrication defects and we want to be able to like easily generate controllers for them. And there's one robot called the Ionocraft, which flies with high voltage ion thrust. But it's very fragile and takes a lot of assembly time. So the goal is to use these methods to fly that robot and that's something that is still pending and kind of slowed down in recent times.

[00:17:03] JM: You also write a blog on some issues around robotics. Tell me about your blogging.

[00:17:10] NL: So my blogging is best summarized as wanting to see how, or like wanting to help the robotics that we want to create. So kind of this vision of perpetual future of robotics that I've mentioned before is let's try to mitigate some of the issues that we're seeing in like machine learning bias and some of the more difficult to tease at issues of digital systems. So digital systems and platforms provide amazing scale. I fully expect some successful robotics companies to do that. But I think that if we're ahead of the game, we could make it so that robots are more like broadly helping people and there's not going to be – It's like the fast food chains are going to get robots first. And if all of the lower income people are interacting with only robots, I want to know like what harm is going to be done there and how to mitigate it, which is kind of how I came to the name of democratizing automation. And it started on robots, but I'm really like any digital system that interfaces with the physical world.

[00:18:16] JM: What's an example of something you're writing about?

[00:18:20] NL: So the thing that I'm currently writing, which is going to come out in late December for whenever this episode is released, is kind of making a statement with why model-based reinforcement learning, which is the subfield of reinforcement learning that has a structured dynamics model in it. So you use supervised learning to learn how the environment evolves is kind of a candidate to eliminate some of the uncertainty and reinforcement learning via like being interpretable with how the actions are selected by an agent. So we don't just have a neural network policy that is a black box computation where you give it a state so you like give it a personal ID on a news feed and then the news feed chooses your next piece of content. With model-based learning, you're going to be able to see like which pieces of content were evaluated and then hopefully get some insight into how the system is working.

And then also with the dynamics model, we're creating some structured learning of the environment. So then we could also use that to kind of create other systems, which there's a whole new field of offline reinforcement learning that's kind of coming into this. But I'm kind of trying to make a statement on like this is something that people should be learning about because the peak performance of model-based reinforced learning is converging to model free and it has some nice properties that make it a little bit more user friendly and in the terms of the engineer making it and then also the person who encounters the product.

I can say that this is pretty related to something that I wrote about last week, which is there was a series of blog posts from Facebook AI that we're detailing what they are trying to do to mitigate hate speech and misinformation on their platforms and what they're trying to do with AI. So they tout some impressive numbers, but it's very hard to read into because the systems are – At least one of the two is an end-to-end reinforcement learning system. So it's very hard to understand what it is doing to the digital platform.

So if you're thinking about reinforcement learning, it's this framework of there's an agent. The agent interacts in the environment and it gets a reward. And in robotics, it's nice because the agent is almost always the robot and the reward function is almost always parameterized by the task. But in a digital system, it's actually not that clear, because you can think about it in different ways. Like the environment might actually be the user and the agent might actually be the news feed. But it's kind of nice to think of like the user as the agent. So you get this little backwards path which makes it harder to design and then the reward function is like advertising dollars. And it's very hard to know how that feedback loop will affect someone's mentals, like how they view the world and how they come back to the app and how they interact with their friends.

Like in the blog post, it's going into like what do they actually say that they're doing. And as a company that's just starting this initiative, it's unsurprising that they don't say very much. They point to a couple papers and then they're like, "We did pretty good at this," but it's not convincing to me. And I like to write about like why are these things not convincing and what would they have to say to do better?

[00:21:36] JM: Well, given that you mentioned Facebook. I'd love to know your perspective more about how the interaction between universities and industry fuels research particularly in the reinforcement learning space.

[00:21:51] NL: Yeah. So I would say this is something I've been starting to think about more and it's like very – I'm right in the middle of it as I have done research with Facebook. I have collaborations with them. I have an internship lined up with Google for next year. So I'm right on this border. And I'd be very happy to say that I've found a group at Berkeley, which is we call ourselves GESE, which is graduates for engaged in societal education or something similar. It's a it's a mouthful. But we have multiple grants investigating kind of the political economy of RL, in reinforcement learning. And for the current state of the like interpretable and safe reinforcement learning area is somewhat fraught, because a lot of the research – Like what would be more like “independent research think tank orgs” are actually funded by industry as well. So I'm not going to name any, but I know I've talked to people that do research at these public think tanks that are trying to create beneficial AI. And there is the whole – Generally, this is grounded in the firing of Timnit Gebru, which I think brought up some underlying issues.

And putting aside that specific incident, it's like the people that I know that work at these partnerships are concerned about what they can say because their funding comes from industry, yet they are trying to be independent. And there's always going to be difficulty there. So there are ideas like what if the big tech companies bankroll, like donate a lot of money to create new institutions. And it's to the point where it's like even if they give enough money for certain number of years of runway, there're still these implicit ties. And the companies are the only ones that have the scale and investment to really do anything without a national push. It's an incentives problem and the scale can be quickly scary. Please give me a point to continue on, because I have a lot on my mind.

[00:23:51] JM: Well, sure. I don't even know the firing of who?

[00:23:56] NL: So, Timnit Gebru, she's a great researcher in the space of AI Ethics and she pioneered a lot of work there. She's also been pioneering a lot of work for inclusion of indifferent affinity groups in the AI community. So the AI community is and historically been extremely challenging to underrepresented groups, which is something that I am troubled by. Like I have a permanent email on my website. If you're underrepresented, please email me if you want to try to work with people, work with me or people I know. And it's like I feel like a minority even doing that. And it's hard because there's so few people and there's getting a side of the whole diversity inclusion problem.

So Timnit was a lead of the AI Ethics research group at Google. So Google, in interest of their PR, which has now blown up a little bit, it had this large functioning ethics group which was doing a lot of the pioneering work in the field and especially from the industry side. I am aware of Google AI had a lot of work and they're restructuring now. Microsoft Research has a big group as well on the ethical or transparent AI and they've been publishing a lot too. And it's kind of this problem of what happened was that they were submitting a new paper for internal review. And that paper got flagged and then Timnit had a series of demands that were not met, which culminated in her being fired is kind of the way that it is viewed now. It’s not complete information as of yet. But with her firing, it raised the question of who can investigate ethical AI systems? And if the companies are the ones funding it, there is a problem that they eventually will draw the line when the research would affect their bottom line.

So the general view is that the paper that was in internal review was commenting on large language models, which months earlier Google released a statement saying that large language models on the new attention-based transformer architectures are used in every Google search. So it was commenting on the potential downsides of bias and energy usage in these large language models, which are now core to Google products.

And somewhere along the line, Google drew the line saying, “I don't know if we want to publish this exactly as it's worded.” And that restriction of academic freedom especially from people that are looking for the best interest of society becomes quickly scary, because this sounds pretty minor, but we might not have even heard of it if there was larger ethical concerns being raised and the private company has a lot of power to kind of quiet these concerns.

I mean, personally, I'm very concerned about it, because I want to apply to jobs at the border of industry and academia. And if I'm working on re enforcement learning and trying to say like, “This is how reinforcement learning may affect people.” There may be companies that don't want to take me on or let me freely speak my work because I'm saying that reinforcement learning or machine learning techniques they're using may have harm on individuals in ways that are hard to measure. And I think being able to speak freely is what we as academics and people are passionate about these problems want as like ground zero of a job offer. That is the point, is to advance and educate the common – Is to advance the science and educate the public and make sure these things benefit society.

[00:27:35] JM: Yeah. What is the correct mechanism for disclosing some kind of discovery of a negative usage? I mean I think what's tough is a lot of these technologies, you may be able to reveal something negative about it, but that's not even actionable. Like what kind of disclosure would be actionable?

[00:27:58] NL: Yeah. Interestingly enough, most of the dialogue I've heard on the space is kind of going on Twitter in an ad hoc way between like various machine learning researchers. And we can talk about a couple of the ideas that I have heard of recently and I think are compelling in different ways. And they tease at the problem in different ways. One of them is starting with like an AI system whistleblower protection type thing, which is I use the term whistleblower intentionally, which is to say if you see something suspicious at your company, there are specific protections for people to be able to document these and send them out privately, which would get the documentation done. It doesn't necessarily mean that a fix would be done. And there's other things, like there are some members of congress that are talking about algorithmic fairness techniques and setting up regulation that I mean I wish it could be worked towards. I don't expect it to be passed anytime soon, is like regulation to call for like certain reporting structures and documenting what data is used and what, so to say, actions are used by the systems.

And a third thing that I think would potentially help remove some of the risks of industry academic ethical research kind of about that trio is a national computation setup where – So part of the reason that industry labs have such draw is because they have way more access to computation and they have access to research engineers that help people pursue their goals, is if the United States or some other group joined together to create a large academic compute structure where academics now no longer have to go to industry to be able to run the same experiments.

So at NeurIPS, which is the biggest machine learning conference this year, the best paper award, which is extremely prestigious, helps people's careers a lot, was awarded to GPT3, which was from Open AI. And the training cost of that was 4.6 million dollars. So that trading cost for one model is well above the cost of running many academic research labs. So that disparity is something where a national research computation center could help rebalance the space, so to say.

Yeah, I think I could tie this back to reinforced learning a bit to try to share some takeaways from earlier conversations on like why reinforcement learning, which is not used in a ton of products yet. So most big companies aren't using reinforcement learning yet. But why it is potentially dangerous and why I'm thinking about it and kind of how this would relate to some of the policy questions and some of the scares of like industry relations is.

So reinforcing learning, I've talked about like agents and environments and it's kind of unclear. And what's something that we're trying to work on is say like what makes certain problem scenarios more appealing to reinforcement learning and what makes them more risky at a societal level of harm and an individual level of harm? So you kind of have these three axes which are like will reinforcement learning work? Are there unknown wide scale risks and are there unknown small scale risks? And it’s kind of like you can think of it like an axes, and once you get into certain areas of this space, you probably want to have stricter regulation on these systems.

So something like the power grid is a canonical example of where reinforcement learning actually has been used on certain scales and works very well. So like famously Google's data centers have their cooling systems optimized with reinforced learning. And why this works well is because we have a very good physical understanding of what is happening. We have like the laws for electro dynamics, and those go into how these cooling systems work and the power flow. And essentially when the reinforced learning system says to do something that is completely unphysical and impossible, the classical systems can take over and stop the reinforcement learning from doing harm. And in that case, the harm would be a higher electric bill or a short temporary outage of a digital service. That's not too hard. Or that's not too hard to wrap your head around just being okay.

But on another end of the spectrum might be using reinforced learning to learn to drive a car. And eventually this car may try to go onto the highway. And we don't know how humans interact with cars very well. It's hard to model. There're whole fields of that in human robot interaction. And like if the reinforcement learning car is to mess up, there could be very immediate, very high harm to both the person in the car and any bystander. And that is something that might want to be limited, is we have no real way of guaranteeing it, but there's a practical idea that reinforcement learning could be used because it's a general framework that can be used to solve a lot of hard problems. And therefore I might not want to use it in that case. And having conversations like this and writing these ideas down I think is going to be very important to kind of how the national like understanding and mood towards these systems progresses.

[00:33:16] JM: Given that you mentioned GPT3. Let's zoom in on GPT3 for a moment. Do you see any potential applications of GPT3 as dangerous?

[00:33:30] NL: In the way of physical harm, less dangerous there. But the way it could become to physical harm is generally through mental harm, which I could see. And generally the idea – And this is an interesting example again. Like Microsoft invested a lot of money in this and the idea could be that they're using it for their products. And most of the flags raised at GPT3 was that it was trained on a dataset that was not representative of how these companies would like to view the world. It was trained on a Reddit corpus, which if anyone has been on Reddit, there's very clearly a different way of speaking and it sometimes can be interesting and not appealing. But the model therefore reflects those patterns. And the machine learning model, fundamentally, they pick up on patterns. And one of the ways that I've heard it described is GPT3 is very good at rearranging words and it can do interesting things when you like give it a prior of a code snippet, it can rearrange code into almost functioning like frontend code, and it can do amazing things. And I think in specific use cases it will be very useful for generating content that is repetitive or in things like video games where the story is recreated every time you log in. That stuff is fantastic and there's lower risk. But if it is interfacing with users in a more crucial environment where a user is interfacing with like a medical doctor and the text is generated by GPT3 for the chatbot that interfaces their reality with the real person doctor. That mediary could be harmful.

And just by subtle biases and kind of accumulated harm that people could pick up by repeatedly using these systems, the word micro-aggression is a terrible term, because even one “micro-aggression” can very heavily change people's lives and views of the world. And especially accumulating similar things can have a very drastic effect on people's mindset. And that's kind of like it depends on how they license this digital product and they are trying to make it a product, which means that it's going to go out into a world somehow.

[00:35:49] JM: Let's return to the subject of robotics and reinforcement learning. Where are we at with consumer applications of robotics? Like I saw there's a pretty exciting product from Amazon recently that I think highlights where we're at. There's like a product that it's like a ring security robot. Like basically it's a robot with a video camera that flies through your house. I think that's a pretty cool application.

You can imagine the end state of this is I think where you have like a robot that kind of flies around with you all day and maybe is sort of like a Google Home with wings attached or with a quadcopter attached. I can imagine it being pretty useful to just have like a little familiar that flies around with you that you can talk to all day. Do you have any perspective on where we are at with physical applications of robotics?

[00:36:43] NL: Yeah. So I kind of ask myself this a lot, and I like the Amazon ring example that you brought up. I actually think that's a great use case because it's fairly limited and I think it's going to do its job. Anyone that's heard a drone take off indoors is going to be – They're very loud. They're very scary. And I think that like the right digital protections can be put in place where it actually physically turns itself off when it's not flying. So people like to joke about these things listening to them in their house and I really don't think any of these companies listen to people through their smart speakers. They can get the information they need by looking at what people click on and the invasion of privacy would be a huge PR blob. So it's like I think that's a good product. I personally don't think I would like having drones following people around just because of the kind of restructuring of the airspace and the restructuring of like what going outside would mean.

But I think it's – Like I go back to the Roomba. The Roomba from iRobot, which is the robotic vacuum, is probably the most successful consumer robot to date. And it's a low enough price point. It does what is expected of it, and it's kind of fun at that. And that like trio is what I think makes robotic products very exciting. It does what you want of it. It's cheap enough and it's exciting. Like that's what makes robots fun. Like the smart speakers, they aren't very exciting, but they're cheap and they do something. And I kind of see a next generation of robots being like smart speakers, but also like simple and able to move around, because there's kind of this like affinity with robots and they're kind of awkward and things like this a lot of times and that makes people like them.

And then there's also the question of like is a nest thermostat that automatically regulates your home environment? Is that a robot? It's like if smart speakers are so close to becoming these little mobile entities, it's like where do we draw the line? And I think it'll be in the continuation of things that have been moving into the home, so the digitizing of the home. Smart speakers is the best example. I think those things are going to take the next step to be mobile before something like a dishwasher unloading robot is possible. It's just there's so many problems that need to be solved to do these physical tasks that are so hard.

[00:39:06] JM: Do you know anything about the most acute problems in getting to the next phase of robotics applications at home? Is it just a series of really small and frustrating tactical milestones we need to reach or is there some grand problem?

[00:39:26] NL: I think it's – I can phrase the grand problem as there's a lack of a grand problem formulation. So I was listening to this panel of a lot of people I respect and some of which I know talking about the limitations of like reinforcement learning and why hasn't reinforcement learning had the breakthrough of imagenet or have the breakthrough of transformers, which is language processing? It's like why haven't there been these structural step function changes that are used in tons of fields? And it's kind of because the data for every problem is just a little bit different and everyone's kind of progressing these individual problems. And this structural – It's harder to like have a big competitive environment to solve the grand problem, because it may not exist.

So like what is the problem in trying to get a home robot to work? Is we wanted to be able to do a ton of different things robustly and safely where like learning to unload the dishwasher is a problem of getting enough training data. And then we're not going to be able to – Like if I trained a robot and had it pick up plates a ton of times. If I put it on a different dishwasher, it might not be able to do it, which is a problem of generalization. Like we're getting there, and I think imagery, like computer vision is going to help that a lot. But it's like every problem is just a little bit different more than just noise in the data. It's like you might get a different number of sensors or a different number of motor variables as well, which just makes it. So like plugging and playing these different methods is really hard.

[00:41:03] JM: Do you know what your work at Google is going to focus on when you get there?

[00:41:08] NL: So it's likely to be a continuation of my work on model-based reinforcement learning, but I can also comment on what I'm aware of the robotics team is working on there. So kind of the robotics team at Google DeepMind is definitely one of their smaller teams I didn't necessarily know it existed until somewhat recently. It's led by Martin Reed Miller. And they're very focused on these things of like modular robots, domain transfer, like actually having robots work in the real world. And there's kind of like this gap between deep reinforcement learning, which is kind of going for end-to-end systems to solve all the problems with data versus practical robots where it's like machine learning is probably most useful and just like how do you translate the images into some data that's useful for a controller? Or how do I get my controller performance to be a little bit better? Add a data-driven system to it.

And I don't have a specific detail on it, but like most DeepMind projects, it's very much on like pursuing a vision of what machine learning can do to kind of transform a field that's not necessarily been stuck in its assumptions, but it takes a very big effort and long-term vision to like make these breakthroughs that are transformative. So that that type of thing is something exciting that could go back to the idea of like why don't we have robots in our homes? It's like these big problems need a big scale to actually address them. Just like DeepMind just solved the – Or I'm not an expert in biology. So I can't say solved. But they made huge progress on protein folding through their reinforcement learning structure and know-how that was developed through solving games. And it's kind of this grand vision of like artificial general control intelligence is I think a term that the robotics group does, which is how do we design systems that can learn to solve every task that is thrown at it? So it's general. Able to control multiple robots to solve a bunch of different tasks.

[00:43:20] JM: We've had a few shows about frameworks for using reinforcement learning. So you went to Berkeley, right?

[00:43:28] NL: Yeah.

[00:43:28] JM: So I'm sure you're familiar with the team working on Ray, right?

[00:43:34] NL: Yeah. I haven't used it, but I think there's a lot of like interface for these very polished digital products with some of the like machine learning problems. And it's like Ray is for auto tuning of parameters, right?

[00:43:48] JM: Well, Ray is a lower level framework. And then there's RLlib on top of it, that I think is for the parameter training.

[00:43:57] NL: Yeah. Specifically, I'm not sure if I've used that one, but there's so many. I mean I have used RLkit, which is from a colleague. I just like started building my own, and it's like Open AI has a library of reinforcement learning algorithms. And there's a lot of different libraries and it seems to be that – I don't know other fields, but reinforcement learning seems very sensitive to like the little details. So there's a famous physics simulator that's called MuJoCo, and it's like if you have version 1.5 versus version 2, you can't reproduce the same results, which is a problem in the field, which is like using a – Or using a parameter tuning, advanced parameter tuning scheduler for like online tuning of an RL problem, which in a paper that's I helped out is under review now. Essentially, it broke the simulator. So eventually tuning the parameters really well broke the reinforcement learning simulator. And that kind of goes back to like the question of the grand problem, is like we need to formulate these things more clearly so that all the libraries can kind of join together on these one problem and like actually compare results.

So like RLlib is one of them and RLkit. There's like Open AI baselines. There's whole other different – There's definitely more that I haven't heard of. So it's like this problem of them being just a little bit different is tricky.

[00:45:30] JM: Why is reproducibility so important? Can we just have all the different solutions asymptote towards the same the same result?

[00:45:39] NL: Reproducibility is very important as a like scientific verification mechanism. So, essentially, the way that results in reinforced learning are very frequently represented is a performance in reward is the word that's used over time graph. And people like to take the median or they show these error bars. And when we're doing a primary digital task, there's something called the random seed, which you set all of your environments to work in the random seed. So you set PyTorch. You set the simulation environments random seed. You set NumPy's random seed and all these things are used at different points.

And if you don't have access to the code, there's no way to very well verify that they didn't manually set the seeds for the ones that worked the best. And that kind of – And when you couple in this like digital uncertainty, which is just numerically there's a lot of variation across simulated trials along with the fact that people want to use these different codebases. There're too many sources of uncertainty to tease at. So like as a researcher, I've reviewed papers. I've looked at people's code. And normally it's hard enough to even run it at all. And then trying to port it to a different library, I've had experiences porting to a different library where you like can't reach the same performance.

So there's sometimes exploitation of these interactions between environment and your codebase that can be beneficial or harmful, and they're extremely hard to document, which is the idea of reproducibility is documenting what those are. And it's a big challenge in reinforcement learning.

[00:47:17] JM: Well, Nathan, as we draw to a close, are there any other topics that you'd like to explore?

[00:47:24] NL: I don't know. As it's a software project podcast, I'm kind of interested in the idea of where is reinforcement learning going to interface on more lives? And some of these things with like software optimization and on the digital side I think are where it's going to explode. So this podcast has covered a great number of topics. I'm not referring to my own episode, but like you get a lot of different concepts here. And I think the appeal of reinforcement learning, and I can end with pointing to offline reinforcement learning, is that all of these datasets, if we do our job well as reinforcement learning researchers, is you can come with a data set and a problem that you want to solve related to that dataset and have algorithms that will improve and give you a way to act with the data.

So I have this vision of like if I log my own health data for a long enough time, I want to be able to have a reinforcement learning algorithm that looks at like my health metrics and my sleep and workout schedule and says, "What are the things that I could fine-tune to feel a little bit better?" Or it's like it's probably going to be used on social media apps. So reinforcing learning, you say, like I want to maximize user engagement. And like maybe TikTok is starting to think about using this or the algorithmic social medias, is like how do we optimize the metric? And we don't really know exactly what the problem space is, but we want to have algorithms that can figure this out.

And, really, why I'm working on robotics is because it's fun, but the idea is that robotics is like a embodied version of all these problems where it's easy to see what happens. Like I was convinced when I saw a quadcopter crash 14 times and then the 15th time it can fly. Like you kind of see these things and then you become convinced and want to see where it can go out into the world. And that's why like DeepMind has been doing very big problems, but also like I think that in a few years it's going to be of people's interests that have digital datasets that they weren't quite sure how to learn how to act from it, that we want to be able to create algorithms that can take that data and return a policy. And offline reinforcement learning is the idea of you are interacting with the offline dataset rather than interacting with the world in an online setting. And it's becoming a big subfield of reinforcement learning. Kind of for that reason of it, can open up our field to a lot of different problems.

[00:49:51] JM: Okay. Well, that sounds like a great place to end on. Thanks for coming on the show, Nathan.

[00:49:56] NL: Yeah, thanks for having me.

[END]