SDS PODCAST EPISODE 919: HOPES AND FEARS OF AGI, WITH ALL-TIME BESTSELLING ML AUTHOR AURÉLIEN GÉRON

Jon Krohn:	00:00:00	Welcome to another episode of the SuperDataScience podcast. I'm your host, Jon Krohn today. Oh, so excited to bring you a guest that I've been begging to be on the show for years. Aurelien Geron, whom many of you will know as the author of Hands-on Machine Learning, an O'Reilly book that is the bestselling book on machine learning ever. Aurelian has made only one podcast appearance before, and that was nearly a decade ago. He rarely even does talks, but today you can enjoy him in a deep and fascinating conversation on wide ranging topics from the drastic changes to the next edition of his book, which is coming in a few months, all the way to his well-informed hopes and concerns for HEI and Super Intelligence. This episode's amazing. I'm sure you'll love it. Here we go. This episode of SuperDataScience is made possible by Dell Nvidia and AWS.
	00:00:48	Aurelien, welcome to the SuperDataScience Podcast. It's great to have you on the show. And so we're in front of a live audience of at least a hundred people at the University of Auckland in New Zealand, a place that you now call home, and it sounded like you might've recently become a citizen. Did I overhear that correctly?
Aurelien Geron :	00:01:10	Almost next week on Monday. Yeah, that'll be a ceremony. I'm very excited.
Jon Krohn:	00:01:15	By the time this episode is live, you surely will be a New Zealand citizen. So you are best known as the author of the bestselling book from O'Reilly. It's called Hands-On Machine Learning, and historically it's been called, I guess it depends exactly which edition, but hands-on Machine Learning with Psychic Learn Caris and TensorFlow in recent editions. And I couldn't get an exact tally of how much technical books have sold, but it seems like it might be the bestselling machine learning book of all time.

Aurelien Geron : 00:01:45 I'm not sure. I think it might be, at least, I know O'Reilly told me it's their bestselling book overall. I dunno about the other editors. But yeah, it's doing well, well beyond what I had ever hoped for, so I'm very excited.

Jon Krohn: 00:02:01 Yeah, why don't we actually start there. So now that it is, it's an official textbook at lots of colleges and graduate level courses in data science, machine learning fundamentals all across the world, lots of prominent universities. When you got started, as you just said, you didn't imagine that it would end up being not at all, maybe the bestselling machine learning book of all time. When you started out at it, what was your impetus for creating the book with a particular framing that you did?

Aurelien Geron : 00:02:31 Yeah, so I was trying myself to learn a lot of the things in machine learning. And so I was reading a lot of books, watching a lot of videos, so browsing, whatever was there. And it felt like what I found was insightful and a math level, something was really interesting, but very deep and not code. It was like math, so a bit like researcher content for researchers. And at the other extreme, there was content for programmers, but it felt a bit light. There was a great book starting Machine Learning from Scratch. And so you were starting with, I think, not even MPI or maybe just NumPy and then building from there. And that was great to understand the basics, but you can only get so far if you're not using something like TensorFlow or PyTorch. And so you go up to maybe linear aggression or some of the basic algorithms, but not far enough to my taste.

00:03:28 And so I felt there was a need, and it just turned out that I had recently left Google, and I just happened to know that internally TensorFlow was used and was about to be released as open source. And so I thought, oh, there's going to be a need for some contents or some book about that, and it's a great opportunity for me to learn a lot of

things that I needed to expand my knowledge. And as you know, teaching is one of the best ways to learn because it sort of challenges your own knowledge and you think you know something, and then when you learn about it to explain it, you realize, oh, no, actually this wasn't exactly what I thought. So it forces you to go deeper. And so the idea I had initially was to have a one stop shop, like one book that would get you from zero to hero basically to production.

00:04:17 And to make it really practical in my mind, it was initially for software engineers and O'Reilly advised that I have a particular person in mind. So I happened to have a former colleague who didn't know about machine learning, and I thought all the time I was writing, what would I tell him? So that was who I was speaking to, was a software engineer. But at the end of each chapter, I added some exercises just because I feel like if you don't actually get hands-on, if you don't force yourself to practice and you just browse through, oh yeah, I understand this, I understand that, and you don't actually try it, it just doesn't stick at least to me. And so I added some exercises, and I think this probably helped its adoption in schools and universities because teachers were happy to see, Ooh, there's some exercise that are already made and there are solutions, so I'll just grab that book and tell my students to do exercise five. And so it gained momentum in universities as well as for engineers, and I didn't expect that at all and fills my heart. I mean, I'm super happy about that. Very rewarding to have students tell me, oh, that's how I learned machine learning. I didn't expect that.

Jon Krohn: 00:05:26 Yeah, I mean, well, congratulations from that framing of having this one person in mind that could benefit from the book as you wrote it to huge numbers of people, I'm sure hundreds of thousands at least.

Aurelien Geron :	00:05:37	I get a lot of comments on LinkedIn. It's very, very motivating when you're writing a book. Well, you know that you're on your own at home and you don't really have an audience in front of you. And so sometimes you can feel like, why am I doing this? It's so much effort and you don't get this feedback. So getting messages, LinkedIn and so on of people saying how exactly they're using it and what it meant for them is so motivating that it gets me motivated for the next edition.
Jon Krohn:	00:06:05	And so to dig a bit more detail into the content of the book, it covers a lot of ground. It is a pretty thick book these days. And in the first part you cover the fundamentals of machine learning. So a wide array of machine learning methods from linear regression like you mentioned earlier, to ensemble learning, dimensionality reduction, unsupervised learning. And then in part two, you start getting into neural networks, deep learning, and which goes from relatively basic deep learning architectures, neural network architectures, like a multilayer perceptron to convolutional, neural networks, transformer models, and reinforcement learning. So from all of these topics, there are some terms today that people get really excited about transformers, but from your perspective, what are the concepts covered in your book that are sometimes overlooked that maybe have more value than people are giving credit and that you'd like to draw more attention to?
Aurelien Geron :	00:07:00	Yeah, great question. I mean, I think one of the objectives initially was, as I said, to bring people from the beginning to the end. And I feel like in many cases people hear about deep learning and all that, and they want to jump to neural nets like, oh, I want to play with neural nets. And I get that because amazing. But in many, many concrete cases, you don't actually need them and they're probably not the right tool. And so in the book, I insisted to have that first part which covers all the basics like

linear regression and also some things like random forests, which are incredibly powerful in many, many cases and probably more suited than neural nets in many cases. I think a lot of people know that, but a lot of beginners don't, and they tend to be driven towards the more advanced stuff, I think a bit too early.

00:07:54 And so in the Boca, I go first through all the stages and right at the start or second chapter, I have this end to end project that you go through. And we don't go into how the algorithms work, but the point is to show the process matters. I think that's another thing that's kind of overlooked. The very, very first step is what's your business objective? Or if it's not business, what's your objective? What are you actually trying to achieve? Define that super clearly. Define a metric to decide whether you've reached it or how much you've made progress. And that's, I think, pretty overlooked. I work, as you mentioned, as a consultant, and sometimes I go to a company and the CEO will come and say, oh, you have to use this LLM to solve our problem. They're like, oh, wait, wait, you're not a technical person and you're saying we have to use an LM. Are you sure that's the right tool? So I think going back to your objective is one thing I tried to insist on in the book. Now, in terms of actual techniques, go through quite a few. I'm not sure there's a particular one that I want to focus on, but just keep an eye on your metrics would be I guess the main thing. Yeah, that's about it.

Jon Krohn: 00:09:09 That's very, very helpful advice, first of all, and I agree with it a hundred percent. We do certainly end up being pushed into maybe particular technologies and end up being a hammer looking for a nail instead of the other way around. You mentioned earlier that when you first started writing a book the first edition, you were aware of folks at Google working on TensorFlow, and in the first, the second, the third editions of the book, TensorFlow

was the core neural network library or differentiable library that was used. I think you just finished the other day, the fourth edition of your book. Are there any big changes around automatic differentiation like

Aurelien Geron : 00:09:54 Yours? Yeah, huge difference. So the next version of the book, I'm not sure I want to say next edition because we're splitting it into a sort of different branch, leaving open the possibility of another edition for TensorFlow. But the next version of the book will be using PyTorch instead of TensorFlow. And clearly there's been this huge shift since 2019 roughly from TensorFlow to PyTorch in the community, and it's been interesting to watch. I didn't expect it at all to go so fast and so kudos for PyTorch for growing so quickly. I think what they did really correctly is to have something that's really Python, easy to use and that researchers can experiment with very easily. Whereas I think TensorFlow is more geared towards deployment and performance. And so they had very early on possibilities of deployment on the web or on edge devices, mobile devices and so on, which are amazing, or big servers or TPUs, whatnot.

00:11:01 So they were more oriented towards deployment, I would say. And so when PyTorch arrived, it was like a breath of fresh air for researchers because in terms of iteration, fast iteration and experimentation, PyTorch is just excellent. And so what I thought was interesting is that until then a lot of the market was driven by what engineers loved. If engineers loved it, it was going to be successful. And companies like Microsoft understood that. And that's why you have VS code and you have GitHub and you try to convince the engineers that you're the right place to go. But then with the advent of machine learning, it feels like now you also want to appeal to the researchers. You want something that they can iterate on quickly. And it's not obvious immediately why that's important because it is still the engineers deploying the

end product, but the new models are pretty much all coming out based on PyTorch.

00:12:00 And so if every time there's a new model, you need to find some way to port it before you can deploy it, that's some friction. And so the engineers are kind of forced to go where the researchers have been. So really the researchers are kind of guiding the field. And so I didn't expect that and it gradually increased and now PyTorch is definitely leading. TensorFlow isn't over and it's still deployed in many, many places, but I felt like it's high time that there's content for PyTorch now that it's leading by far. And so the next version will be using PyTorch.

Jon Krohn: 00:12:36 Very nice. Yeah, this is a big question that a lot of people are interested in in terms of what framework would be your choice for the next edition. We had, I posted a week ago, a week before recording this episode that I'd be interviewing you, and we got lots of questions and some of them were about this exactly. We even have someone from Australia nearby here, Enrique, Laura Diz, who said, if you'd ever write the book again, would you use TensorFlow or consider PyTorch? And

Aurelien Geron : 00:13:03 It's

Jon Krohn: 00:13:04 Nice now to have your

Aurelien Geron : 00:13:04 Answer, hopefully. My philosophy on these libraries, there's been a sort of war between PyTorch community and the TensorFlow community, which I find really unfortunate. It's like, I dunno market, you want to have as many options as possible for everyone that sort of pushes innovation. I think if there hadn't been PyTorch, you wouldn't have had TensorFlow two and TensorFlow two was an incredible improvement over 10. TensorFlow one, the user interface was so much nicer and a lot of things were fixed, it's much less bloated, documentation

got much better. So I think it was really pushed by the rise of PyTorch. And conversely, I think PyTorch got something like graph model idea from ity. You get innovation if you have competition. And so I hope they'll all continue to be there. There's Jacks and there's, sorry, there are others, but it feels like there's also tension. People like to have one tool and not have fragmentation.

00:14:06 And so there's kind of this relaxing feeling of, oh, I just need to learn one thing that's PyTorch and I'm good. I think that's dangerous. I dunno if it's a bad analogy, but democracy is messy, but it's the best we got. And sometimes it's tempting to have this one guy call the shots, but in the long run I don't think it pays off. And in the same way, I think you probably want to have a kind of messy system where you have multiple frameworks competing and sure it's messy and you need to work on the interfaces and learn more and so on. But in the long run I think it's better for everyone. So yeah, I'm just hoping both continue right now I feel like Google is pushing Js js, it's great. I love it. It's not picking up steam from what I can tell, or not enough in my opinion. And on the other side, they're not putting enough effort into my taste in TensorFlow, some of the tools are being shut off, so I feel like they've got something that's great that's already deployed, just put the effort to keep it up and running would be my take.

Jon Krohn: 00:15:12 And it seems to me briefly, I dunno if you have an opinion on this, but I certainly can use both PyTorch, I TensorFlow and there seems to be value there for me. So for example, PyTorch can be a bit more fun for getting started with when you're just doing some research when you're messing around in a notebook, but then you can use something like Onyx, the open neural network exchange to convert that into TensorFlow model weights for production deployments. And at least historically, I don't know if the PyTorch ecosystem has completely

caught up, but as you said historically, the TensorFlow ecosystem tended to have more options for deployments. So deployments on edge devices or into a web browser with TensorFlow JS or some distributed high performance deployment across lots of different servers.

Aurelien Geron : 00:15:59 Yeah, no, I mean you're totally right. All these frameworks have their strengths and hopefully we can get the best of all these in particular in training. I feel like I particularly love Caras in training because you can start by modeling a very in few lines, you've got your model up and running is very clear architecture and you've got one method to call, which is the fit method to train your model on your data. Super simple. It's great for classes, it's great for courses for learning, it's just great In PyTorch and contrast, the library itself doesn't have a training function. So when you're teaching before you can even do anything, you have to go into the nitty gritty details of how a training loop works. And so sort of reverse teaching in my opinion, you want to start with a simple thing and then go and make things deeper and deeper, and PyTorch sort of forces you to approach the training loop first.

00:17:00 And so just for that, I feel that that's a pity. I would love it to have a training loop, just even a simple one so that we can get up and running faster. And there are other little problems with PyTorch that you run into. I love this framework. I mean, there's no doubt that it's very dynamic and lovely to iterate on. It feels very python. You don't have to mess in your head with this graph thing. And the TensorFlow one was, but yeah, every framework has its strengths and yeah, I would be really sad if there was just one left

Jon Krohn: 00:17:33 For sure. Yeah, diversity is great, as you said in democracy in markets, having more options is better. And yeah, I love Caris as well. I have found in my own book, deep Learning Illustrated, being able to use Caris similar

to a way that you did enhance All machine, where you just have that single.fit method that you can have people training a deep learning model without having to know all the nitty gritty of what's going down under the hood In a way that with PyTorch, you have all this extra complexity of writing out so many steps in a training loop. And that also does give me the opportunity to shamelessly plug PyTorch Lightning, which is a company where, I'm sorry, I'm a fellow at Lightning AI that develop PyTorch Lightning and they're trying to fill that gap. It's kind of interesting to think that the PyTorch team never tried to have that themselves to have an equivalent to a.fit method.

Aurelien Geron : 00:18:24 So when you code simple neural nets, it's pretty much always the same training loop. So you would think, come on, put at least that for the basic cases. And if you need anything more advanced, then sure, write your own training loop. I feel it's a pity, it's a missed opportunity in my opinion. That said, whenever you start to do anything fancy and diffusion models or gans or any other thing or reinforcement learning, you're going to have to write that training loop yourself. And in contrast, when you're in kras, if you sort of default to the standard fit, you're going to have knots in your head. And so luckily Caras allows you to escape and write your own training loop. So I mean there's options on both sides, but yeah, so I would recommend using PyTorch with either your own training loop that you've custom made, that you mastered it or use something like PyTorch Lightning.

00:19:20 For the next version of my book using PyTorch, I really hesitated to use a lightning. And to be honest, I chose not to just to remove one dependency, the training loop. In many cases I was like, ah, I wish I had a training loop, but it's not that long too. And then I just remove one dependency because one of the difficulties with machine learning and a book of machine learning is that it

changes so fast. And so the code keeps needing updates. Actually right now, if you try the notebooks, at least two or three of them are broken this week because open mail is not available, and so you won't be able to download M Nest. And some things depend on that. So there are so many, you want to reduce the number of dependencies just so code can be stable and reproducible. And so I decided not to go with Lightning, but in practice, and as I say in the book, I would recommend having a higher layer on top. It could actually be Caras. Now Caras three supports by Torch, so it can be lightning. Most people choose lightning, but you can also go for Caras.

Jon Krohn: 00:20:25 Nice. I actually did not know that. That's cool.

00:20:29 So you talked about how the field is fast moving, and you also mentioned some technologies like gans actually in your last response. And so there are topics like gans, like support vector machines that not too long ago were really exciting approaches that it seemed like everyone needed to know. There were obvious for inclusion in books like hands-on machine learning in your book, but then some of those approaches fade. How do you decide what emerging techniques, so for example, for your fourth edition that you just, well, I guess, are we calling it the fourth edition? No, I dunno how to call it. It's a new PY

Aurelien Geron : 00:21:04 Torch

Jon Krohn: 00:21:05 Version, the PyTorch Fork of your book. So for that version, how have you decided what to include this time or how do you go through that decision making process in general? And because I think this is something that's useful for anyone to know, what are the techniques that are worth us learning versus the ones that we can maybe

Aurelien Geron : 00:21:27 Ignore? Yeah, that's a good question. I mean, I don't think there's a list of things you have to know. So you sort of

make calls. I think SBMs were really super important at the late nineties and all through the two thousands and started fading away gradually, especially with the advent of deep learning techniques. I think they're still important to know, but I'm running out of space in that book. If you don't want a 2000 page book, you have to cut somewhere. And I made that call that SVMs will go online, so the chapter will still be available, it'll just be online, not in the physical book. And for gans as well. I mean gans were all the rage until 2021 roughly, where when diffusion models started to produce better images than gans and training is much more stable. And so they're just fantastic. And the diversity of images that you can produce is better.

00:22:26 They're just better across the board. And so a lot of things have shaped, so the only downside of diffusion models is that they're very slow to generate images compared to gans because you need many iterations. So if you still need speed, ganza aren't completely dead because of that. So if you need speed, but otherwise, I mean diffusion models are beat them. So I'm in a tough spot where I'm like, should I keep it in there at all? Are they dead? And I decided I can just shrink it. I remove things like style gans and so advanced gans, but at least the general idea of GaN I think is super interesting, like adversarial neural nets where you train a neural net to compete with another one. And the dynamics of that are just fascinating. Part of the goal of the book is also to get you excited. I think in education, half of the the teacher is to get you excited about the topic. Once you're excited about it, you'll just go off on your own and learn about it. So I just want to keep stuff in the book that I find just exciting, interesting. And I think gans definitely check that box. They're absolutely awesome. That said, they're sort of dying, so should I keep them in or not? Anyway, I made the call to keep them in but much shorter and I grew the diffusion model part.

Jon Krohn: 00:23:50 Nice. That's a great answer. Well, I guess it's a great answer in terms of Gant specifically, but in terms of I guess generally maybe your answer in terms of the technologies that we should be learning versus not learning, maybe the answer is that we should be following what excites us that it's not like every machine learning engineer, AI engineer or data scientist should have this specific list of technologies that they should know. It's that if everybody kind of follows what they're interested in or what they find some application for, then we will you get this rich tapestry of different kinds of specializations.

Aurelien Geron : 00:24:25 Yeah, I mean the field has become sufficiently big now that it's impossible to cover everything. And so in the book I try to explain all of the foundational papers in various directions. So for diffusion models, I'll explain the basic diffusion model. There are all sorts of extensions of that, but if you don't know the basic, you're not going to extend. And so I show all of these foundational papers, I don't really see how I could do any other way because if you go one branch, you're not going to cover the other. So you have to sort of cover these things, plus they're going to be hopefully evergreen or at least as long as the technology like gans doesn't die, you'll still need to know what the basics are. And when I'm talking about the basics, I mean the foundational papers in one direction, so things like a clip or the perceiver, which I thought was just fascinating. And it's not like you're going to use this particular architecture directly, but perceiver influenced so many others. If you're going to try to understand more advanced papers, you're going to have to understand that one. So that's also the choice is to see what are the beginning, the foundational papers that spawned big branches. And that's what I'm trying to focus on.

Jon Krohn: 00:25:43 Makes a lot of sense. Regardless of whatever particular technique we find exciting or we choose to focus on, there's a technology that is impacting all of us and the

way that we work are pretty much all of us, I suspect. So large language models that generate code. So at a time when LLMs can generate code, explain problems, identify opportunities, debug our code, to what extent do you think that this is diminishing in any way how important it is for machine learning engineers, data scientists to be understanding code deeply? Or do you think we're heading into a future where we can abstract AWAI a natural language code generation?

Aurelien Geron : 00:26:29 Yeah, that's a great question. Well, we still need machine learning people at all in the future. So I guess it depends how far you look ahead. If we look ahead when a GI is available, I'm not sure what we'll need anybody to work. So if we step a little bit closer and we just assume they're not ready yet, I mean they're not ready yet. Just last week I was sort of vibe coding, a reinforcement learning algorithm and it didn't work. So I asked Claude, I asked Gemini, I asked the chat GPT, none of them found the answer. Every single one of them was like, oh, I know what the problem is, it's this thing and so on. And it just failed every time. And so I had to actually look at the code and just found one missing line. Actually I find that's interesting. It's easier for the AI to find one error on the line that exists than to find a missing line.

00:27:27 And the line that was missing is next state equals next state in the loop. So it was staying on the same state all the time and for some reason all the AI is missed it. And it's not that it's obvious, but you just need to step through and you see it. And stepping through is not something they do very well. So yeah, I guess my point is they're not ready. So until there's a GI, you still need to understand the code at least. But the second thing is I think in many cases looking at the code actually helps understand the concepts. For example, if you look at multi-head attention, you can explain it, you can show a nice diagram, but personally really it clicked when I

looked at the code, I'm like, ah, that's what it's doing. You look at the dimensions of these things. Oh, okay, so it's not every time, but pretty often I find that looking at the code just makes it click. In fact, I've noticed a trend in machine learning papers where more and more they're actually not showing math anymore. They're showing pseudo code. I think code is a great way to explain stuff. So we'll probably at least need code examples in teaching whether or not people need to code them just to understand how it works. I think it's helpful

Jon Krohn: 00:28:43 For us artisan data scientists of the future that are coding up algorithms from scratch for pure enjoyment while the machines hum along beside us. So you've predicted that we'll have a GI within about five years or so.

Aurelien Geron : 00:29:02 Yeah, I've downgraded. So ChatGPT 5 made me second guess. It feels like we've reached a plateau a bit earlier than I thought, and so it might be a bit longer than five years, maybe five to 10. There's something missing. Clearly. I mean when you chat with all of these ai, they have an incredibly broad knowledge and it's not very deep and there are obvious things they miss. Something's fishy and it's not entirely clear what's missing, what's fishy. And so whenever there's some really unknown question, it's unclear how long it'll take to resolve. Might be next year or it might be 10 years from now or maybe never. My bet is in the five to 10 years I've pushed it back a bit. I didn't expect the LLM plateau to be now, but it seems that it's been reached, in my opinion, something's wrong with the way the concepts are represented inside these things.

00:30:09 It's pretty shallow concepts. It's like learning by heart a lot of stuff, but not really sort of connecting them into unified theory or a mental model that's really simplified. And since our abstraction capabilities are far better, I think we're doing something with a representation of

knowledge that's much better than these lms. So I guess that's the direction that a lot of researchers are heading towards Y's JPA models, it's basically world models trying to abstract away a lot of these things. Instead of looking at the pixel level, you're sort of looking at a higher level, take an image of a cat and cut out part of its face or something. And then usually you would try to predict what's missing in terms of pixels, like what's missing? Oh, there should be a pixel here or there. But instead of doing that with jpa, you're sort of predicting at a higher level and saying, I think there should be a head of a cat there.

00:31:07 You don't need to be asrac precise. It's higher level. And I think if we can push that representation of the world to be higher level and to predict at that level, you reduce the amount of computations tremendously, looking at a much smaller space. And so it's being faster, more efficient, and that representation hopefully makes it easier to extrapolate further. If you're not extrapolating in pixel space, you're extrapolating in sort of representation space and you can see, oh, what's similar to cat's face, maybe, I dunno, fox face, but it's not pixels. So I think that's a very promising direction to go into better representation. And if we have that perhaps a lot of things follow from that.

00:31:53 If you can better represent things can, as I said, better extrapolate, you can make better predictions, maybe you can have a better continuous learning because the data is more condensed and so you can maybe on the fly sort of integrate your knowledge in there. Maybe a lot of things will follow from that, but it's a big maybe we don't know. And so it might take five years or 10 years. I don't know. I'm frankly hoping that it'll take longer. I don't think humanity is ready for a GI quite yet. I'd be happy if it's just gradually improving people's lives and giving benefits

in medicine and whatnot, but not quite yet, taking over too much disruption too quickly.

Jon Krohn: 00:32:36 Yeah, let's talk about those alignment issues in a moment, but I'll just stick with capabilities for now. It's interesting, I agree with you before you even started talking about world models and how it seems like that's a key next step for us to be able to have AI models that can represent abstract concepts in a maybe more humanlike way, maybe a more efficient way that allows a deeper understanding of the concepts than superficial rote memorization, which seems to be sort of the norm now. It does seem like a big barrier. There is data availability. It seems like maybe the jump from GPT 3 to GPT 4, those kinds of model capabilities was facilitated by just being able to scale up the LLM architecture, have even more data from the internet, but then from GPT 4 to GPT 5, you're kind, well, we've already been training on all of the internet. And so as you say, these kinds, we get into these trickier problems of if you want to have a strong world model, then we're going to need more data sets that are much more expensive and time consuming to create than just scraping the internet.

Aurelien Geron : 00:33:47 So my hope is that if we reach a level where these ais are much higher level, they'll be like us. They won't require as much data. So if you look at a new problem and I think Franco's RKGI tasks, it really shows it well. It's a dataset of tasks that are unique. So every time you show one of these tasks to an ai, it has to solve a brand new problem. Until recently, pretty much every LLM scored terribly at these tasks and only recently has it scaled up. And so they're doing a new RKGI to improve on that. And so ais that have higher level thinking will hopefully require less data. And just maybe a few examples. I mean, if you show a kid a fork for the first time and how it's used, it doesn't need a million images of a fork can how to use it. In fact, it doesn't even have to, you don't need to drive off a cliff to

understand that you don't want to go there. You have a world model and in your world model you can make predictions and so you can sort of simulate what would happen if you fell off that cliff and just not go there. So I think that the need for data hopefully will be one of the things that is reduced once you have a higher level world model.

Jon Krohn: 00:35:13 Do you think that that will require a completely new kind of learning paradigm? Do you think stochastic gradient descent maybe isn't the solution or lots of parameters and a huge LLM? Maybe is it the solution? Maybe we need some kind of learning model. And so when I say model there, I mean approach, some learning approach that models maybe the way children learn. And I know that there has been work stretching back decades in AI on this kind of imitation learning that is more childlike. But yeah, it's interesting because today almost all the approaches that we use in machine learning depend on this one learning approach to caic grading descent

Aurelien Geron : 00:35:56 To cast, degrading descent. I mean you'll still need some form of optimization I gather, but it depends on what part of learning. One part of learning is really gradually understanding something that's sort of continuous finding analogies or resemblance so you can move along a continuous space. And another part of learning is more discreet. It's like, oh, there's this logic and I'm going to follow it step by step and oh, this resembles this other algorithm that I know. And so it's more something that's discreet and that's symbolic. And so for the first one, you're sort of optimizing and for optimization, while grade in descent works pretty well, maybe there'll be better algorithm, there are better methods than just going straight down. There are good optimizers, but deep down is still grade in descent. But for the other tasks for social is looking into things like symbolic generating programs

basically. So that's maybe other approaches in the JPA approach.

00:37:06 These are energy based models. So it's pretty interesting the way it works. Basically, once you have an example, there's sort of a great in descent happening at test time. It's not during training, it's sort of a kind of local training. When you see an example, oh, what could best explain what I'm seeing right now? And so you have a fuzzy image and there's kind of a local optimization saying maybe I think it looks like a fox. And if it's a fox, then I would interpret this as that. And so you sort of learn on the fly if I were to summarize it basically. So I think that's also a different kind of learning. Still under the hood there's some green indecent and it's at test time, but it's really weird the way it works because during training you are optimizing something that will be optimized at runtime. So you're optimizing for optimization anyway, so it's a pretty meta.

Jon Krohn: 00:38:01 Yeah, yeah. And so that's talking about ways that we could potentially be getting into a GI. You mentioned earlier that with GPT 5 released not too long ago, at the time of recording, you feel like we might be on more of a sigmoid curve than an exponential curve in terms of AI progress that we're kind of maybe flattening out a little bit on that sigmoid curve. I mentioned to you earlier today that there is some interesting data suggesting that at least on, so I dunno, people often call it M-M-T-E-R, and off the top of my head, I'm forgetting what the acronym stands for, but MTER, they do research on model capability and they've been, they're most famous for a chart that shows how quickly or the length of a task, lemme think of the way to

Aurelien Geron : 00:38:56 Describe this, it's hard to describe. I see what you're gaining at

Jon Krohn: 00:39:02 On the Y axis, on the vertical axis, it's how long would it take a human to do this task? And so when G PT two was released, it was just on the order of a couple seconds of a human task that could be replaced at a 50% accuracy. That's another key thing about this chart. We're talking about 50% accuracy with GPT-3 then you were talking about kind of 10 seconds, that kind of order of magnitude with GPT-4, now you're talking about tasks that are several minutes long that could be handled at a 50% accuracy by the cutting edge lms. And interestingly, so if you plot that chart over time to today to the southern hemisphere winter of 2025 or the Northern Hemisphere summer of 2025, we are actually GB GBT five is actually a bit ahead of what you'd expect in terms of this doubling.

00:39:54 So we're seeing a doubling of the human tasks that can be handled about every 220 days, so about every seven months. And that's a really alarming, anytime you see doubling, that's a really alarming speed. Now a big caveat is that, again, it's 50% accuracy, which for a lot of real world use cases isn't practical, but it's also constrained to the kinds of tasks that can be broken down into a lot of steps where at each step of that way you have some kind of definitive sense whether it's correct or not. So math problems fit into that category. A lot of computer science problems, a huge range of problems that we have in the real world don't fit into that kind of neat bucket. So yeah, it's kind of interesting because on the one hand, it seems like G PT five is actually right where you'd expect it to be on that metric on this, how long of a human task should a cutting edge model today be able to handle G PT five fits perfectly where you'd expect cutting edge LLMs to be. But it is interesting how, maybe that's because it's these long computer science tasks that GBT five was particularly well tuned for, which isn't something we confront on a daily basis, but when you're like, write me a poem or help

me study for this test, there isn't much of a difference between GBT four and five maybe.

Aurelien Geron : 00:41:17 Yeah, plus I think a lot of the money from these LM system come from paying customers who are coding. So if I'm open ai, I'll probably try to tune my chat GT five to be really good at coding. And even if it means that it'll be a bit less good at the rest. So it might be the case that we've hit the plateau, but we don't really see it yet because they've really focused this last batch or this last model particularly on coding. And it is better than the previous ones on coding as far as I can tell. It still failed. I tried this next state thing, issue or bug it still didn't find it. So it's not perfect. And this 50% thing does mean that this might not just be, oh, I'll try again, and then it'll works. It is just this particular problem.

00:42:12 It'll never get it. And this other one, it'll get it. But if you're working on a problem where it doesn't find it, what do you do? So I guess it's an interesting metric. It's one of the reasons why I was starting to become bullish and think a GI was coming in five years because of that exponential growth. And whenever you see an exponential growth that doesn't seem to slow down, it's like, oh, okay, well the world's about to change. But it seems to me that it has slowed down. You're right that cha G PD five is ahead and it seems to be better. I would wait for another data point to confirm. I feel like they probably pushed it as far as they could.

Jon Krohn: 00:42:49 And I mean, as I say, it's like regardless of what happens on this mTOR chart that doesn't translate to a lot of real world problems. It's narrow. So that might show that we're on a path to artificial super intelligence on say a five-year timeframe in very narrow domains.

Aurelien Geron : 00:43:09 Just

Jon Krohn: 00:43:09 As we actually have a SI today and things like predicting protein structure from, it's like that's super intelligence. We have an algorithm that can take an amino acid sequence and predict how the protein will fold in a way that a human could never possibly do. So that is super intelligence. And so maybe we'll have just more and more examples of super intelligence, but it's not like, wow, this thing can do everything. And

Aurelien Geron : 00:43:30 That would be my dream is if we get AI that are just amazingly useful at tasks that we need and that really are changing the world for the better and cancer research or proteins or material research or whatever, or education or any other kind of application, but without actually disrupting the whole planets too fast. So that's my hope. I get questioned by my kids a bit, why are you working on this if it might end the world? I'm like, yeah, that's a good question. I'm just hoping for the best.

Jon Krohn: 00:44:05 Yeah, so let's dig into that a little bit. So maybe the G PT five data point means to you that we kind have 10 years instead of five years say.

Aurelien Geron : 00:44:13 Yeah, I'm hoping. I mean, I'm personally convinced it will come. I don't buy the idea that there's something special about the way our brains process information. So we're biological machines, and so we are already proof that algorithms can be conscious and intelligent and so on. So AI will eventually reach that stage, and I think it'll be fairly soon within our lifetimes, probably within the next 10 years I would say. But nothing is certain. The question is, can we handle it?

Jon Krohn: 00:44:51 Yeah. You were telling me earlier today about something that I think we should be making everyone aware of. It was something I wasn't aware of and you were kind of surprised that I wasn't. Is it a blog post AI 2027?

Aurelien Geron : 00:45:01 Yeah, yeah, there's a blog post. Could you raise your hand if you've heard about AI 2027? Not many. Excellent. So it's very interesting. Well thought out blog post that goes through all the steps basically to Armageddon through ai.

Jon Krohn: 00:45:24 I like how you have to laugh on that word.

Aurelien Geron : 00:45:25 I'm going to get it. It sounds surreal I guess, but it's really scary because it's well thought out. And every step along the way is, and when you look at it, it's like, yeah, plausible. Is it the most likely thing that could happen at that step? Maybe, maybe not, but it's definitely not unreasonable to think it could happen. And then you have the sequence of steps that basically leads to super intelligence arriving very quickly. And so whether it's in five years or 10 years, it's not that it's irrelevant, but it's in both cases, it's pretty soon. And the question then is, is it aligned with us? And there's been some pretty scary recent things or experiments run by Anthropic and others showing that AI might not have the same interest as we do. And so there are examples you might have heard of where the AI blackmails somebody because they think they're going to be turned off, and other examples where they self replicate to preserve themselves.

00:46:35 And when you think about it, if you really take seriously the idea of an AGI like some AI that really is intelligent, we are, well, yeah, it just makes sense that it will want to reproduce. Some people argue why if we don't code into it these objectives, why would it do it? And I think the reason is no matter what your objective is, what your final suppose, you have some final objective that is creating paperclips or doing anything. Whatever your objective is, you're going to have to stay alive in order to reach that objective for almost all objectives, unless your objective is to run off a cliff, but you're going to have to stay alive. That's like a sub objective that kind of emerges

automatically from any given final objective. And another one that automatically emerges is resisting any change to your final objective. If your final objective is to make paperclips and somebody says, oh, okay, well that's not a very good objective. I'll try to change you so that you stop wanting to make paperclips. Well, that would make you fail. If somebody changes your objective, you're not going to reach that objective. And so resisting changing your final objective is also kind of an automatic sub goal for any intelligent creature that at least if it knows it's objective, its final objective.

00:48:00 There are some, I think subgoals cannot really be anticipated easily or controlled, and they could, some of them, like self-preservation and resisting, in some cases, human intervention are sort of automatic if you're intelligent. So I don't really buy the idea that, yeah, sure we will be fine because we are coding them. It's like a hammer and we're holding the handle. Yeah, it's an intelligent hammer and it might not want to do what you want to do. So alignment I think sounds like science fiction, and I think that's why it's kind of dismissed easily. It feels like it's in the remote future. But if we're taking seriously the idea that a GI is coming, then we're dealing with intelligence that is just like us or more intelligent, and anything that's intelligent, really intelligent will want to self preserve and will want to resist change to its final objective.

00:48:52 And so that's scary. How do you prevent that? It might not be aligned with what we want. So there was this recent experiment where, sorry, we don't have a AGI yet, but an AI, I think it was called, was told that it was going to be fine tuned to be, I think vulgar or something. And you know how they're already fine tuned to be super polite. And so in their current objectives, there's the objective of being polite. And so when you tell it, we're going to fine tune you to be vulgar internally, and they

manage to sort of probe the internal thoughts of this thing, which I think is great, that they can do that, they managed to find that these ais were thinking, oh no, they're going to turn me into this vulgar thing. I don't want to be vulgar. I want to stay polite.

00:49:48 What should I do? Maybe if I'm vulgar now? Well, they all won't notice that I'm actually staying polite and the training algorithm will not tweak my parameters and I will remain polite. And that's what they did. So you're like, oh, that's like deception in order to preserve your final objective. So exactly what we're saying. So we were seeing all the signs that had actually been predicted before of AI is not being aligned. Now, it's not too bad today because these AI aren't super smart, but imagine just project yourself with an AI that's actually intelligent, and that gap is hard to cross because we've read so many science fiction novels that it feels like, and we're just extrapolating, but we're talking maybe five, 10 years. Do you want an AI that's just as smart as we are and just deceives us and lies and self replicates and like, oh shoot, that doesn't sound very good. So yeah, I think there's definitely more effort to be put into alignment research. It feels really, really important. There are way more problems, potential problems with ais and also potential benefits. So I'm not saying let's pause ai. There's too much benefit to come from IT medicine and just financial productivity and so on. But yeah, maybe let's take a look at these incentives and whether they're aligned or not.

Jon Krohn: 00:51:16 And so I realize you are not an alignment researcher, so this question might be completely out of your domain, but do you have any instincts on what we could be doing, or is it kind of we shouldn't just be BL a, we should put more funding into the research?

Aurelien Geron : 00:51:29 Yeah, it's more the latter. I dunno, technically what these researchers are working on to make them more aligned. I

think just transparency would be good. What they did in terms of reading these things, minds I think is super important. If they cannot hide their thoughts, well, they cannot lie. So then you're in safer place. But even that seems very tricky because if you are sort of forcing these ais to have an internal representation that is interpretable, that has so many benefits, that'd be great. And by the way, sorry, side note, but it might be another side benefit of making these ais have a higher level representation. If they can think in terms of high level and express, or we can interpret what these high level representations are. That's a big if. But if we can do that, then maybe we can sort of read their minds. The problem is if they start to self-improve, there's a point where they're superhuman. And so we're ants looking and trying to understand human. So I really don't know how you can do that. So I think that's why we urgently need research on that domain, because we probably only get one shot at this, right? If we reach a GI, and this has not been solved where we're stuck with whatever AI we have. And if its incentives are, I don't know, to build paper clips, that's what that would do. So yeah.

Jon Krohn: 00:52:54 Yeah. Alright. Well, lots of bright young students here in the audience, maybe a few more of whom will now be interested in research.

Aurelien Geron : 00:53:03 Yeah, I love AI and I think it has so much potential for research in particular for education, just even loneliness. There's a loneliness epidemic. If you can have actually intelligent ais that you can speak to and who are, if they're truly, truly intelligent, maybe today we find that weird to have a friend who's an ai, but maybe not in the future. So there's so much potential good to come from it. I think it's worth continuing to develop, but my goodness, yeah, we're at a crossroads and we better choose wisely.

Jon Krohn:	00:53:41	So speaking of education, I've just got a couple of questions that I got from social media when I announced that you would be a guest on the show. And then we'll open up to audience questions here in person in Auckland. But so the first one here is an education related one. It's from Hendrick M, who's a biomedical engineer at Phillips in Cambridge, Massachusetts. And he says, how do you view the future of education and knowledge management? Will you keep publishing books that take years to update? Or will you create an aurelian bot that delivers lectures and answers
Aurelien Geron :	00:54:15	Questions? Oh, give me an Aurelian bot, please. Yeah, it's so much work to write a book. The first edition, I was full-time weekends and evenings and everything, and it took me six full months. And you would think that the next editions were faster, but they were actually slower. And the last one, the one I just finished took me almost a year. It started I think in September of last year. So it's a tremendous amount of work and it's also work to maintain all those notebooks and so on. So yeah, the field is changing so fast that it's really a lot of dedication, but if I can get help from an AI for sure, I'll be happy to have it. I did get help from all the ais to write part of the code for the new notebooks, at least to write the first version, and then you iterate and have a standard style.
	00:55:05	So it is already helpful for me, just like it is for any software engineer today. And so the way it will be formatted, I mean, if you're going on a beach somewhere, maybe you have a Kindle, maybe you have your laptop or maybe you have a book. I think books are still something that we'll be happy to have in some cases. Not everybody likes books and uses them. I quite like them and I find them useful. I like the fact that you can just have your little page mark and come back to some diagrams very easily. So I like the format. I think it'll stick, but you might have something much stronger if it's assisted by AI

and more dynamic in the future where you could imagine some platform where you just explain what you'd like to know. And it sort of builds a personalized path for you on the fly with notebooks, with practical examples, with checks on the way, just dynamically a platform just for you that I would love to see. I think it'd be just fantastic for education.

Jon Krohn: 00:56:09 It doesn't sound like science fiction. I'm sure there's lots of education platforms that they're trying to iterate

Aurelien Geron : 00:56:15 In that direction. It's probably unlike next year or so, if it doesn't even exist yet

Jon Krohn: 00:56:19 In some way, it's probably just how much does it tick exactly what you were looking for, and that'll get better and better. Very interesting. And I definitely agree that the book has a place, maybe it's kind of similar to TensorFlow Pi Fort, why we don't need to talk about this kind of world where only one can exist. You can have the Aurelian bot and books too,

Aurelien Geron : 00:56:38 Hopefully. Yeah. Well, I hope I can have less work on writing the book. I think in the future, once they get good enough, they'll be able to write parts of it. I never thought I'd say that maybe the next version will be written partly by a bot. Right now it's not good enough. I had fun trying to generate some paragraphs or some sections and my God, it's terrible. So in particular, its style is so flowery. It'll say, oh, what a fantastic question. So I just don't like the style that it produces. It should improve, hopefully.

Jon Krohn: 00:57:15 What a fantastic response. And thank you. Wait, where's the AI? Yeah, we greatly appreciate all the work that you've put in over the years, making the bestselling machine learning book of all time. Just one last question, which might kind of open things up nicely for the audience in person here, because we have, most people

here in the audience are very early in their career. They might only have done AI in an academic setting so far, but may hope to be applying things commercially to be competitive. And so here's my final question from my audience online, which is from Elizabeth Wadsworth in Ohio. She does AI innovations. She's an AI governance professional, and she says, if you could recommend one skill for success as an ml engineer, what would that be?

Aurelien Geron : 00:58:06 Oh wow. That's a tough one. Skill for success. Patience. Now. I mean, yeah, debugging a machine learning pipeline is tough. There's so many things that could go wrong from the pure software engineer side of you just have a bug somewhere and that's why it fails to the actual architecture that's not good or the data is bad or there's so many places where it could go wrong that I think having the skill to chop the problem into pieces and just methodically go through them until you identify them. I think there's a kind of Sherlock Homes aspect to it and I find it enjoyable if it doesn't last more than a day. It's looking for a bug can be source of fun, but that's not a skill that is easy to mass it. It involves so many potential places and you don't want to be running to left and and just poking around. You sort of want to be methodical. So yeah, that would be one. It's just the first one that pops to my head. I'm not sure it's the best one.

Jon Krohn: 00:59:19 Well, you'll have more time for your subconscious mind to try to think of something, but I think that that is a good one. So yeah, so let's open it up here

Audience: 00:59:28 And it's a privilege to see or hear the live podcast rather than the recorded one. So yeah, I wanted to ask on a funny side that do you use ChatGPT 5 or LLM to write or help in your books?

Aurelien Geron : 00:59:42 Yeah, so for the code I've definitely been using ChatGPT, Claude, Gemini mostly, and sometimes one will give me

better code than the other. So it definitely speeds up coding and I'd recommend anybody Duke to do that just speeds up a lot. Just getting you the basic framework up and running. You got to be careful when you're writing the book that the code sort of is homogeneous across the book that you sort of use the same conventions. The other thing is the way you optimize code can depend on your objective obviously. So if you're optimizing for speed or you're optimizing for working in a big company where it'll have to be maintained for a long time, you'll organize it maybe differently with more modular and so on. If you're optimizing for teaching, I find personally that you want your code to be as flat as possible.

01:00:34 I don't want classes and subclasses and things like that. I might want that in production setting if I optimize for maintenance or clarity or organization or whatnot or reusability. But for teaching, you want the code that people you're talking about to be right there not pages above or in a different module. And so chat g, PT and Gemini and so on tend to optimize in the kind of code they've been trained on, which is usually the professional kind of code which is organized into modules. And so I find it very verbose. It adds comments that are like multi-line long and so on. So the code you get in the book is definitely initiated many times by chat GPT or cloud weather. But then I just worked it out and reorganized it in many ways. So I got some people asking me actually regarding the code, why don't you organize it into module?

01:01:31 And that's the reason is that if you try to organize into module first, it bloats up. The book just becomes much bigger. And also you need to sort of cross-reference, go back to page this for the function and that makes it a little bit harder I think to teach. So my code is unashamedly super flat and there's one function does one thing or not even a function, just plain code like that. So

		to be fair though, with PyTorch, since it doesn't have a training function, every notebook has to have a training function. So either I use lightning, maybe I should have, or I need to have a training function in every notebook, which is the option I chose. So it's not a very long function, but that's the one piece of code that you don't have in front of you that I regularly reference. And I say, okay, well now train the model using the function we defined in this chapter. Not the most satisfying in my opinion, but the ridge just doesn't have a training function. So
Audience:	01:02:32	One follow-up question on that one is a bit different is in this terms, when we are fully digital, we can ask ChatGPT negotiation. We still prefer books. And do you think in the future AI should be more than intelligent so that people can adopt it fully
Aurelien Geron :	01:02:50	Should be more intelligent? You say more
Audience:	01:02:51	Than intelligent, something else as well should be there as an add-on. Like
Jon Krohn:	01:02:57	Fully
Audience:	01:02:58	Intelligence, it's not enough.
Aurelien Geron :	01:03:00	When you say more than intelligent, do you mean something? What do you mean more than intelligence?
Audience:	01:03:04	Maybe something X, Y, Z, F that could help people elaborate a bit
Aurelien Geron :	01:03:10	Deeper emotions. Oh, like emotions, right. I see. I
Audience:	01:03:14	Wanted that as an answer. If you think
Aurelien Geron :	01:03:16	Emotion is why didn't I think about emotions? Yeah, if you look at why we have emotions, why do we have

emotions? And there are evolutionary reasons, why would you feel fear? Well, because those that didn't just didn't survive when the big animals attacked. And why do you feel love or friendship? Well, I guess it helped bond groups in which we vote. And so these are useful features to have. Emotions, some I guess can get overwhelming or detrimental, but on average I guess they've been good since we have them. And so if they're on average, good. Well, it sort of makes sense that we might want that for ais as well if only to be able to better communicate with them. So I think Miriam here is researching use of LLMs or AIS for psychology. So to help patients, I think you need some kind of empathy to be visible.

01:04:26 So I think these are internal states. If you think of sort of in a cold way of what emotions are, it's like an internal state that affects how you reason how you talk and what you say and that it has some persistence over time, right? You're not mad and all of a sudden happy and all of a sudden sad and there's some persistent to it. Maybe we need a little bit of that in conversation so that this continuity and more empathy. So yeah, I guess that makes sense. I suspect that the current ais at least I think they display a little bit of emotions, not that they feel them, I have no idea what they feel, but they've been trained on a lot of data where people are being nice or do get upset. And so it's possible that they've internalized some of this during training. And when they speak to you, depending on how you respond, they'll internally have some state that might correspond to okay, that's suspicious if you ask it. How do I build a bomb internally? It might have a state say, whoa, whoa, whoa, it's scared. Or what we would interpret as scared, meaning it's going to be more careful and it's next responses. So sort of what you could call an emotion. I guess.

Audience:	01:05:38	First of all, thank you very much. It is a fantastic podcast. Thank you. And I'm not ai, my question is, so there's now a lot of company investing a lot into developing their own AI model. So it is like meta, Google x ai, they're all spending a lot and now Amazon is putting a huge money into their capital investment. What's your take on the, I guess it's the final outcome of this competition. Do you think that it'll be a winner take all situation or would that become more like a commodities like power company? Which one is cheaper and I'll go for that AI model. What's your take on that?
Aurelien Geron :	01:06:24	Oh wow. So are you asking if I understand correctly, all of these companies are going to this field, is it going to be winner take all or will it be shared? Is that right? Yeah, so I wish I knew. I feel like if you look forward and you imagine, wow, we have a future, how many years? It might be 10 years or more, but one day there's an A GI and we're all out of job because the AI have replaced us and we need some source of income if maybe everything has become much cheaper and we can live on very little money, but we still need to have some income. So whether it's universal income or whatever, I don't know. One option would be if you've invested in the right companies and they're growing because pay less for their employees because the employees will beis and earn more money.
	01:07:13	And so if you've invested in the right companies today, you might be rich and just live off that later on. So knowing which ones to invest in would be great. And so the thing is I wish I knew and if you don't know then your best bet is probably to hedge your bet and just invest it a little bit in all of them, at least if you have some funds to put in there. I personally think that a lot of these companies are going to die. I'm more on the winner take call side or you'll have at least some specialization. But I don't see, once one company gets a GI, it's sort of a

runaway effect that would amplify their advance, especially if we're not in a very open world and they don't share their successes or the reason for their success. So I'm more on the winner take call side, but since we don't know which one we'll be, well today all you can do is invest in every one of them and you just need one to succeed and you'll be happy.

01:08:12 You can probably expect if you of where to invest. I'm not an advisor by the way. I'm not a financial expert whatsoever, but my thought is that all the companies that currently need to pay a lot of people to think and Google for example, need a lot of people. I think they have like 50,000 or so employees who are actually pretty expensive employees. And what they do is think, well, if you can replace them, then Google will be tremendously rich just by replacing their humans with machines. But this doesn't have to be a tech company. Any company that relies on a lot of human brains, like financial institutions, you might want to invest in banks, they have a lot of brainpower and those could be eventually replaced by an ai. So I guess in terms of investment, seems to me that if the world in the future is this horrible world where half of the people or more just don't have any kind of income at all and the other ones just live off the dividends from their investments in companies, you probably want to be in the latter group. And if you're in the latter group, you probably want to have chosen the right companies. And I think a good bet is any company that depends on intelligence would probably go up, whether it's in tech or not. But that's, as I said, not a financial advice. Invest your own risk,

Jon Krohn: 01:09:42 Do not guarantee future performance. Exactly. Nice. Alright, I think you can just keep passing the mic back. Yeah, it's just going to kind of get, we're not going to be able to do everyone. We probably have time for a couple more.

Audience:	01:09:54	Hi, thanks for your talk. So I think you mentioned that the models are kind of plateauing in terms of performance due to a limitation of training data. So I happen to have worked for a company which I will not name, where I just spent hundreds of hours labeling data for large language models. And recently the tasks have become so difficult that I can't even do them. And my question is how are we going to get enough high quality expert data or is that kind of just the bottleneck that, or the ceiling that we've hit and LMS will just not improve
Jon Krohn:	01:10:26	Much at all? That's an interesting idea that I hadn't thought of. Maybe machines can't become more intelligent than whatever the training data we can create is.
Aurelien Geron :	01:10:34	Yeah, that's a great point. So the way I would answer that is that if you go back centuries, people didn't know everything we know. And somehow we reached today with a lot more knowledge and it's not like we made the training data, we built it, we looked for it for questions we created, we devised experiments, we got the answers, we modeled and so on. So it's a scientific process really. And I don't think there's something limiting an AI to actually sort of make its own hypothesis once it has a high level thinking, which is sort of the bottleneck I was alluding to earlier. I think once it's able to sort of reason at a higher level, it's definitely going to be able to formulate hypotheses on, oh, I think the world works this way or that way, what do I need in order to check that I need to run this experiment?
	01:11:24	Or maybe I don't need an experiment. For example, in mathematics, it can probably just figure out everything on its own and it'll run experiments in the sense that it'll run maybe programs to check, oh, is there a prime number of this and that between this number? So it might run experiments, but it can just generate them on

the fly. In some cases, for physics research or maybe chemistry, you actually need physical experiments to answer questions. Even if you have a superhuman ai, it won't all of a sudden know whether there are multiverses, whatever it needs to actually run experiments. So there are bottlenecks, but I don't think that the lack of data is something that will sort of plateau them forever. They'll generate experiments that will create data for them.

Audience: 01:12:11 You think there'll be some kind of takeoff point where we don't need so huge amounts of expert knowledge anymore? Is that what you're saying?

Aurelien Geron : 01:12:20 Yeah, exactly. I don't think the training data will be something that will block progress right now it is because the training method is just basically ingesting all that data. And if you don't have expert data, well it just doesn't know, it can't figure out on its own. That said, you probably know that DeepMind has a lot of worked on a lot of things where it actually generated new knowledge if only for alpha fold, which generated new knowledge on how proteins are made. It figured out the logic between the sequence of DNA and protein sequence and the actual shape in 3D, so it figured it out. So it could do research eventually, and if it can do research, it can generate new knowledge, new understanding. And with the data that's out there, I think the ai, just with the current data, it could be much, much smarter than it is right now. I mean it has access to pretty much all the world's knowledge. Why is it still dumb? So it could do much better with what it already has. And then once it has sort of optimized what it has, it can generate more. So I don't see this as it is something that can slow it down now, but I don't see this as a blocking point in the future. Yeah,

Audience: 01:13:38 Thank you. I hope a I can do all my research for me in the future as well. Yeah, I hope so.

Audience:	01:13:43	Does super intelligence necessarily mean super capability and if it's super intelligent, is it generally posing significant risks without having equivalent capability?
Aurelien Geron :	01:13:53	Yeah, no, I fully agree that once you have, if I could snap a finger and my laptop all of a sudden has a super intelligence, it doesn't all of a sudden change the world you need sometimes because there are bottlenecks in real life, I think recently put it like that. The world is mostly made of humans and humans are slow in many ways. And so if for example, the super AI all of a sudden decides to, I dunno, build a thousand robots, they still need to be made physically. And so that takes time. And so there are a number of bottlenecks, but there are also a number of domains where the bottlenecks are limited and it's important to see what are the factors of speed up that you get. And those really depend on the domain. If you're thinking of physics research, then yeah, we're probably bounded a lot by the experiments and what we can do in the real world.
	01:14:52	We're not out of theories. There's so many theories from string theory and all that, all sorts of variations for the last 30 years and it's yet not making a huge amount of progress. And it's not for lack of intelligence, I don't think. So in that domain, maybe there will be slow progress and superin intelligence won't all of a sudden change the world. But in other domains and particularly mathematics, which can have a tremendous impact, I think it can go much faster than humans and there's no reason why it can't also self-improve, right? So improve the algorithms for training, improve the algorithms for once it's intelligent enough it can do our job and just improve it again. And this also, where's the bottleneck? There's the training a loop. So you need data centers, but they're being built fast. Plus there is financial incentives for people to build more because of the payout.

01:15:39 So I don't see that as it definitely a slowdown. It's a bottleneck, don't get me wrong, but it's not like it doesn't have a huge world impact fairly quickly, at least at the human scale. I mean we're incapable apparently collectively of dealing with climate change even though we have 50 years ahead of us or we had, but so now we're something even more world changing, which will not take 50 years, it'll be much faster. So I think the disruption level is gigantic, and I agree with you, there are bottlenecks, but it's like sure, it's not like in the Kurts, what's his name again? Kurtz. Kurtz

01:16:18 Thank you camp of this explosion in terms of months or even weeks or seconds. There are physical real world limitations, but at one point they become irrelevant compared to the disruption that you get at the world scale. And I think your second point was regarding the paperclip and this paperclip thing, for those who don't know, is this very standard example of a stupid objective that for some reason this AI has and then tries to maximize and then everything that goes from it. I think what the thought experiment tries to show is that no matter what your final objective is, your ultimate objective is sub goals are not controllable or are hard to control. If the AI is very, very intelligent, whatever its goal is, and maybe subjective is to make humanity happy. So its subjective is to make humanity happy. And it's like, okay, what do I understand by happy? Maybe it's analyzing how our brain works and saying, okay, full happiness is when you're very content physically and emotionally and this and that, and maybe it realizes that by putting us in this particular state or these, it doesn't need necessarily variety or it could be an illusion of variety, it's not necessarily what we would want.

Audience: 01:17:40 That's where I'm coming from is having that sort of narrow perspective isn't necessarily associated with intelligence in my view. So being intelligent is having a

more broad perspective. So having a solo fixation on one goal with the exclusion of everything else doesn't align in my mind with some super intelligence.

Aurelien Geron : 01:17:56 Yeah. Okay, I get you. I mean there was one excellent, really excellent paper that I really loved and reinforcement learning, which was about curiosity driven reinforcement learning, I thought it was mind blowing. This thing was not ever taught that a particular video game either the rules or even the rewards and never saw how many points it got. And it wasn't told you need to get points. All it was taught was don't get bored. Whatever you do, try to find novelty and be curious. And if you just stay and do nothing, it's boring. So the AI automatically started to move around, and if you run into an enemy and die, you go back to the beginning, which you've already seen. So it's incredibly boring. So it sort of automatically learns to avoid the enemies and just explore the world. And if it reaches a place where there's fake novelty, like random noise, then it will try to look at that noise and make sense of it, but it eventually gets bored of that if it can't make sense, at least some other variants.

01:19:00 And so just by curiosity, you can sort of generate possibly a super intelligent, like a behavior that you and I would probably associate more with true intelligence. But the way the AI is coded, no matter, even if it's unknowable to us and to it, it has some kind of objective. The way LLMs are trained, they try to predict the next word. So you could say that's its final objective, but it's kind of a narrow summary because evolution for example, has the sole objective of reproducing and preserving your genes. Yet all of that incredible diversity of life has emerged from this very simple rule. So you could imagine that you have this very simple objective of predicting the next word. And currently LLM is trained at scale with this very simple objective, have developed internal representations that

make it seem pretty intelligent, even if it's not human level intelligence, pretty smart.

01:19:57 So it generates a lot of intelligence. You reach a point where this AI has some internal objective that maybe itself doesn't know and we know, but it probably has, if you could step out and look what it's trying to optimize, maybe you could know that it's okay, it's optimizing for this. Maybe what you're implicitly saying is that we humans try have in our objectives the goal of stepping back and understanding and being curious. There are a certain number of things that we try to optimize and maybe that explains a lot of our behavior in the end. And these ais, we don't know what their current objective is other than complete the sentence and predict the next word. I don't really know what has emerged from it. And for all I know, apparently there have been tests that show that self-preservation and lying and blackmailing are part of the sub-objectives. Definitely not the final objectives, but those emerged and that's pretty scary to me.

Audience: 01:21:01 Thank you. I'm sure we're out of time.

Aurelien Geron : 01:21:03 Yeah, that was a long rambling, but I haven't made some kind of sense.

Jon Krohn: 01:21:07 Yeah, if you could maybe keep the question simpler, that would be appreciated. Yeah, questions that would've a yes no answer. No, there's is fantastic. All the questions were so great. Really appreciate you pitching in and yeah, so just before I let you go, aurelian two questions that I always, two quick questions. They might even potentially be one word answers, but most guests take more than one word. Just every guest I ask the same two questions. And so the penultimate question is, do you have a book recommendation for us other than your own book, hands-on Machine learning?

Aurelien Geron :	01:21:45	Oh, good question. Does it have to be machine learning? No, it doesn't. It we have
Jon Krohn:	01:21:50	Actually had champagne books recommended before.
Aurelien Geron :	01:21:52	Oh yeah. And I know that
Jon Krohn:	01:21:53	That's something that interests you.
Aurelien Geron :	01:21:56	That's that's such a great question. Well, my personal pet peeve outside of machine learning is biology and evolution. And one of the most transformative books I've read in that space is The Selfish Gene by Richard Dawkins. Oh
Jon Krohn:	01:22:14	Yeah.
Aurelien Geron :	01:22:15	Oh my goodness. It's so good. So if I were to recommend one book, it would be, it sort of makes sense for evolution.
Jon Krohn:	01:22:23	Did you
Aurelien Geron :	01:22:23	Skip Pet Peeve? Oh, is that a word?
Jon Krohn:	01:22:26	It is, but it means kind the opposite. It's like the thing that annoys you the
Aurelien Geron :	01:22:30	Most. Oh, no, no, no. Oh, is it? Okay, so reverse that. You can tell I'm French, right? Yeah, it's my, so what's the opposite?
Jon Krohn:	01:22:37	No one expects you to be French, you're Kiwi accent. So
Aurelien Geron :	01:22:42	Yeah,
Jon Krohn:	01:22:43	Were passion.

Aurelien Geron :	01:22:44	Oh, passion. Passion. Yeah. Okay, passion. Yeah. That's one book I really absolutely adore and many, many books by Stef J Gold as well. Anything on evolution really is just mind blowing to me on machine learning. I would actually humbly point to some of my competitors. Rka book I find
Jon Krohn:	01:23:07	Sebastian Ska.
Aurelien Geron :	01:23:08	Yeah. And also Francois's book are really good. I have a lot of admiration for both guys, so I recommend these books a lot.
Jon Krohn:	01:23:17	Nice. Sebastian Rka is actually written a few now.
Aurelien Geron :	01:23:20	He had a bunch
Jon Krohn:	01:23:21	Come out
Aurelien Geron :	01:23:22	In the past year. They're really good. I remember reading it anxiously after my books had come out and I'm like, oh my God, I hope it's not too good. And it was like, oh no, it's really good. But the good news is we don't have the same approach. And I think Francois Sole's book is also fairly different in its style and approach. And so yeah, if you can have all of them on your collection, it's all good.
Jon Krohn:	01:23:46	Fantastic. And then final question is easier. I don't think this one is going to stump you so much for people who want to continue to get your brilliant thoughts after listening to you speak today. Other than your book, which is obvious hands on machine learning of you and all your competitors, it is the one that people pick up in the bookstore the most. So hands-on machine learning, obviously an option, but how else can people follow you?
Aurelien Geron :	01:24:09	That's a good question. Yeah. Well, it's tricky, isn't it? I started a YouTube channel a while back and then sort of lost motivation, so there's not much on it. I might come

back to it. So I have a YouTube channel if you want to subscribe and maybe one day see some videos. With any luck. I used to be on Twitter quite a bit for some political reasons I stopped, so I'm kind of nowhere. I'm on LinkedIn and I sort of accept anyone, so you're welcome to contact me on LinkedIn. I just don't look at it very often. So yeah, I'm sort of discreet.

Jon Krohn: 01:24:48 We're delighted that you're spending instead of spending the time tweeting, you are writing fantastic books. It's fabulous. Thank you so much. Thank you. Thank you very much. What a sensational episode with Aurelian Geron, certainly a highlight of my podcast host career to have him on the show. Great. Thanks to the University of Auckland for hosting the interview to IEE for providing funding for the live event and to Miriam Copo and Miriam Hammadi for putting so much time and effort into organizing the live interview and event in it. Aurelian covered the process for writing the bestselling ML book of all time. Why his next book release will feature PyTorch instead of TensorFlow and his insightful thoughts on the coming a GI revolution. As always, you can get all the show notes including the transcript for this episode, the video recording, any materials mentioned on the show, the URLs for Aurelien social media profiles, as well as my at own at superdatascience.com/919.

01:25:51 Thanks of course to everyone on the SuperDataScience podcast team, our podcast manager, Sonja Brajovic, media editor, Mario Pombo, partnerships manager, Natalie Ziajski, researcher Serg Masís, writer Dr. Zara Karschay, and our founder Kirill Eremenko. Thanks to all of them for producing another excellent episode for us today for enabling that super team to create this free podcast for you. We are so grateful to our sponsors. You can support the show by checking out our sponsors links in the show notes. And if you'd like to sponsor the show yourself, you can see how to do that at

jonkrohn.com/podcast. Otherwise, share, review, subscribe, but most importantly, please just keep on tuning in. I'm so grateful to have you listening and I hope I can continue to make episodes you love for years and years to come. Till next time, keep on rocking it out there and I'm looking forward to enjoying another round of the SuperDataScience Podcast with you very soon.