AI and You

Transcript

Guest: Richard Ahlfeld

Episode 102

First Aired: Monday, May 30, 2022

Hi; welcome to episode 102. My guest today is Richard Ahlfeld, founder and CEO of Monolith AI, and this is really cool for me because he is applying AI to aerospace engineering, and has worked on, among other things, modeling NASA's Space Launch System as part of his PhD, which he received from Imperial College London. So where does AI come in? There are a lot of variables involved in testing machines like airplanes or turbine fans to make sure they work and are safe, and a lot of cases to consider that digital simulation often isn't good enough for. You've got big moving parts in complex shapes with supersonic airflows, turbulence, and so many variables that you end up with what's called the curse of dimensionality: if you want to be able to extract patterns from the data in all those dimensions, you have to gather samples that have representative ranges in all those dimensions, and that means the number of samples has to go up as the factorial of the number of dimensions, which is a function that increases so fast that it's no wonder that mathematicians show it as an exclamation mark. Let's find out more in the interview with Richard Ahlfeld.

Richard, Ahlfeld, welcome to AI and You.

Peter, great to be here.

So, I'm fascinated by what you're doing here, which is the application of artificial intelligence to engineering design, if I got that, right, because it seems to have the potential for short circuiting a lot of the delay and risk in manufacturing and design of new components that need engineering, your thesis and work has gotten into aero-dynamics and an even rocketry and so I like to understand how that application of artificial intelligence helps engineering component and structure design here. There're three kinds of artificial intelligence machine learning we've got supervised learning, unsupervised learning, reinforcement learning, maybe we could start by looking at where we get to, or which ones of those we get to.

Fantastic. So, within the context of what we do at Monolith, I often prefer the term "self-learning models of artificial intelligence." Because if you call what we do artificial intelligence, people who don't work in the engineering research space get a little bit confused about, so where is the AI now, because essentially, what we do is we take in data from the engineering product development process and then we try to recognize patterns in that data that the engineer hasn't seen yet. We try and create models from that data that essentially allows engineers to model things that are so difficult to model that they haven't been able to model them yet and then we're trying to optimize the performance of the system, in a little bit of a sense that it becomes a recommender system and a recommendation of what would be the best performance, and all of that ideally, like without specifically having to program it. So, monolith in the machine learning sense, is a mixture of supervised and unsupervised learning, but is essentially all about creating problem creating problems. Let me try that again and it's all about creating solutions to engineering problems by building self-learning models.

And I'd like to go back to your PhD because this is a very mathematically dense, physics dense, work on the application of machine learning techniques to design and at that point, were you contemplating the future in academia? Because a lot of PhDs stay there. Or did you intend to move out into the private sector already?

So, I loved being in academia, because I've done always been a scientist, I've been experimenting with everything my entire life and I had the best time as a PhD student and researcher, I mean, I got invited to work on the Space Launch System, and NASA, like I was having the best time in my life. So, I love research. The thing is, I also always fancied myself as an entrepreneur, I wanted to do stuff and build stuff, but I also wanted to see impact people's lives and so I gradually moved away from theoretical research, to practical research to robotics research, I wanted it to be closer and closer to what actually happens in the world and I had this one moment as a researcher, where I presented rare events analysis on Rolls Royce turbines, at a big conference in Cambridge, and somebody from Rolls Royce stands up and be like, "Richard, you've made the assumption that you can prevent aircraft crashes, really well, if we give you 6 million engine tests, do you realize that we do three per engine, not 6 million?" And I'd realized I built the perfect mathematical model for a scenario that was never going to be useful and that I think, is the point where research is great, but industrial research and working in a company could potentially be significantly more useful and then I think I gradually shifted over into more commercial industrial research.

So, how do you make up the remaining 5,999,997 tests?

In this particular part of my research, you just don't. I had built a Black Swan model like something that really looks at the most extreme and rare events and tries to model them accurately, which works in finance, because you can connect a lot of data about global markets from millions of people and I'd adapted that to turbine engine without taking into account that you might need more data than an actual engineer working on an actual turbine might have in their day to day life. And so I took that lesson learned and switched what I did to, okay, we need to work with what engineers actually do on a daily basis, how much data they actually have, what problems they actually solve and so, I basically jumped out of the ivory tower into a different area of machine learning research.

So, is the problem in engineering design - and we're talking here things like engine turbine,s and radio antennas, and rocket structures, is the problem in that design too much data or too little data?

Both really just depends on which one, sometimes I have too much data. Sometimes I have too little data, sometimes I have too much data that doesn't mean anything and sometimes I have too little data that means too much. I think as a general condition, the situation I'm engineering is that I have, most of the time, not enough data about what I'm trying to model. It's usually sparse problems, gas data science. So, it's usually little information on most engineering problems.

However, there are quite a few problems, web engineers end up collecting a lot of data and that is because if you sort of work in engineering today, your boss and all of the resellers in the market have been telling you for years, “you don’t need to test anything, just build a model and then you can use this model to virtually design the perfect thing and it’ll just without any data collection just work.” That was supposed to happen already. But it hasn’t happened yet and, in my experience, and I’ve only been doing this for 10 years, this sort of product development keeps being pushed down the road. Because what I found in industry is that when I actually want to develop a completely radically new product, a new inverter, a new Hyperloop, then this contains physics that I don’t understand yet. There is a lot of detail here that nobody really knows what’s going on and the simulation models, the design models that I’ve built up through existing toolboxes don’t capture that yet. And so the cutting edge of what actually happens in engineering is never - at least now it’s not virtual, it is people building prototypes, conducting experiments and collecting data. And these people tend to collect a lot of data, because they will run the Hyperloop tests for days and hours at a time. It’s the sort of hadron collider problem. I don’t know what’s going on. I don’t know what those particles are up to. So, I ended up creating a huge amount of data, because I need to understand what’s going on there and then I’m looking for the needle in the haystack to really grasp the physics and the understanding. And this is I think, a typical big data problem where engineers need machine learning to get it to get two things: one to understand what the system is doing, and two, to model what the system is doing.

Let’s see if I can bring this to an example that will make it concrete, because I’m following for instance, the SpaceX Starship launch and preparation and their tests, obviously, three of them blew up and so that illustrates you can’t model everything you have to you have to test, and I think we’d all like this to happen faster. Well, I’d certainly like to see that launching already. So, anything that makes that cycle faster, is something I’m very interested in and so would be Elon Musk. At what point in that lifecycle of engineering can you make a difference? Is it in interpreting the data from live tests? Or is it in directing how they model, because a test of that is pretty expensive.

So, doing data analytics on things that are already operational when the product is completely built, like Elon Musk’s Starship is a pretty old story. People have been doing data analytics on these kinds of things for ages from predictive maintenance to system calibration and all of these different things that’s quite mature.

Then does that mean that by the time, that’s on the launch pad ready to be tested that the design has already happened and now the only failures are a component not doing what it was supposed to?

Potentially. So, for me, that is usually too late. Where I’m interested in improving the use of data science and modelling techniques is much earlier in the process. I’m thinking that information of how these systems actually perform, and what they actually do, doesn’t need to travel upstream in to what the engineers were originally designing. When on the Starship, they were sort of looking at the fuel tanks and how they generate thrust, they won’t have put those things directly on the rocket, they will have them on, like smaller test stands in different scenarios and they’ve been testing them for days on end to understand really what's going on here. And these are the scenarios when you can try and better model and understand what's actually going on here and that's where I like to sort of come in and say, "okay, we've done two days of testing on your rocket thruster. Do you understand what's going on in different scenarios to see how the temperatures relate how the thrust behaves in different scenarios? Do you see the vibrations?" Yes, I'm trying to pick up more, I'm trying to learn more about what's exactly happening in this test scenario, by picking the data apart with different algorithms and then I love to take this sort of data to train a model that understands exactly how the engine works, and predict how it's going to behave in very different other scenarios. My sort of mantra is, test less learn more, what I'm essentially doing is a little bit I think the most published example is Google DeepMind. They Created a model from a protein database, which could predict which amino acid sequences lead to different types of proteins that totally impressed and kind of shocked the scientific community, because people had been trying to understand and model that using non-AI, but physical methods for, I don't know, decades, if not longer, and the ability to sort of plug in DeepMind's, and Google's machine learning tools to understand what was going on there and to create models that really understood this behavior suddenly allowed them to make predictions of which specific amino acid sequences on a lead to what protein without having to go to the lab and test it and that is what I like doing for the rocket thruster in the early stages, build a model for the early prototype, understand really what's going on and then predict, how is it going to behave and react in scenarios that I don't have the time and money to test today, with a goal that when it then goes to the very final stage, I'm already done, and it's not going to blow up.

So, is your modelling, is it in essence, learning the physics of the situation and in such a way that now your testing can be more targeted, because you don't have to test all kinds of random scenarios that could never happen in the real world.

So, for me the biggest contribution what machine learning can currently do for the modern engineer is that you here have a tool at your disposal, and most engineers don't realize that, they can model ridiculously complicated stuff that nobody can model, from the most complicated turbulence equations, to electronics equations, to rocket fasts, I have something where I can model incredible complexity as long as I can collect enough data about it. And data collection with test stands in the sort of thing is feasible. We've been doing this for a long time. And once I've been able to collect enough data to create a model, I can use this model to avoid doing further testing. So, the funny thing is, Peter, it's a very strange departure from what engineers have been taught in university courses and by the engineering companies for the last 20 years. The sort of common opinion is, you do the design in CAD digital, you do a digital simulation using partial differential equations, differential equations, and physical principles to understand as much as possible in the virtual world what's going on and once you've understood everything there is to understand using equations that you've already purchased from Zeeman and ANSYS. Then you go towards the test stand and that has just literally to tweak and tune, what you've already decided is the perfect solution. Whereas in my experience, that doesn't quite work when I've seen automotive companies try that the final prototype was wrong, it failed faster, and was worse than actually going through more testing early on. Because while a lot of our simulation tools are really good already, like predicting what an Airbus aircraft does at cruise flight, we can model that super well. But there's a lot of things here that we can't, like trying to model how an aircraft lands with the ground effects and the turbulence that's going on there is ridiculously complicated again. And so realizing that there are simply things in the product development process that we cannot model yet and that there is an easy solution where I just collect data on a simplified test stand, create a data driven machine learning model to model that, and then use this model to understand what's going on and make predictions for what's going on in different scenarios is a cheat trick. So, it's a little hack to the engineering development process that can add massive value. To give you a example of something that has just come out last week from the Global Director of Technology at Honeywell, who's been working on ultrasound, flow meters and gas meters and they've tried to sort of follow the standard process of, I design this thing, I run a couple of physical simulations then I test; problem was that the CFD simulations, the physical simulations were completely inaccurate, they couldn't figure out what was going on in reality for this new type of flow meter. With this, they basically then just built a monolith, a machine learning model, based on the test and data got incredibly good results could predict how the system would behave in the most extreme scenarios and basically could optimize and calibrate the meter much faster, much early in design process. So, it's moving away from conventional thinking. But you can solve much harder problems just like Google did, with their DeepMind protein folding predictions.

It's making me think of an analogy. Tell me how close this is. So, I'm thinking of analogies with AlphaGo, for instance, and what it did in the game of Go and coming up with a move that humans would not have come up with in 10,000 games and that was because it was able to exercise a larger space of possible moves than humans were used to doing. We had had a lot of thinking about go over hundreds or thousands of years, but it comes down to were choosing from these kinds of strategies. And so then, what it's making me think is that in designing something and engineering, you can't afford to open up your possibilities to the entire spectrum of physics - making asymmetrical chassis designs, for instance - because it would just take you too long, you have to make an incremental change to the way that it's been done before, if you want to get out the door in time. But is what you're describing that the AI is able to, the machine learning the modelling, is able to consider more possibilities than engineers can afford to.

Yes, definitely. So, one of the sorts of core things that makes this a good trick is that you can consider more possibilities. The one thing that one does have to consider and does have to bear in mind is that machine learning tools that are supervised in their learning approach. So, Google Go right, is a reinforcement algorithm. So, here's something that has been playing through different combinations that no human has ever looked at and is therefore opening up the design space. When we do machine learning based on test data, I can't really leave the design space. It's the classic echo chamber argument, and so I can make predictions for things that the engineer potentially hasn't looked at today. But they have to be within the design space that I've looked at for a rocket. So, it's a small difference, because in both cases, the idea is that I can do machine learning to learn more and expand my horizon. But I think the sort of slight difference between like what you described as Google Go and what I would use for an engineering problem is that for me, I would use the machine learning algorithm. Okay, so there's sort of two different scenarios. Like in the Google Go scenario, I have something that plays against itself tries out different moves and tries millions of different moves, it only gets taught the rules of the game, it plays with the rules and then learns things that nobody's ever had, the time to look at. My scenario is slightly different, because I'm going to be looking at a test stand, where something is constantly changing and I'm essentially just trying to understand how the system works. And mine is complex for the human engineer looking at this, because there's a lot of parameters changing the result. It's a little bit like when you change, when you're trying to set the temperature on your shower, you're going to go there, and set the temperature of I'm turning on the hot water too hot, okay, a little bit of extra cold water - okay, I live in the UK, right? Like, we have two separate taps and you play with the hot and the cold water until you have the perfect temperature. In a rocket test, you will have 100 of those little levers and you will have to play with them all at the same time. A normal engineer will play with one and see what happens, play with the other and see what happens and quickly, using this method, comes to the results that they want to have. A machine learning algorithm is going to be twiddling all of those 100 dials at the same time; it doesn't mind that you get a very complex answer, because mathematical models don't mind that you're changing 100 things at the same time, because they can still learn from but I'm not going to get 100 different changes at the same time and in this way, I can put, what would take me two months of learning, I can squeeze that into two hours of learning for machine learning algorithm and then I can use those two hours to help a human understand what would have happened in two months, because then the machine learning algorithm can pick it apart again and show me, Richard, if you change this little dial, this is what's going to happen to the temperature. But the algorithm does have to learn first what happens in all of the different scenarios and if I've never played with the 101st dial, one that I've never touched before, then the algorithm is too stupid to understand what's going to happen. So, if I've never tested my rocket in Siberia, and I'm putting into Siberia first thing tomorrow morning, then the algorithm I've built for this won't be capable of dealing with it and that's an important factor for safety in engineering. Because this means that I can't guarantee that for Starship, I will be able to prevent any disaster. Unless I've already included anything that could go wrong. In my original design of experiment. Anything that's outside my black box is impossible to model.

Right, and we've already learned what happens when people don't take into account cold temperatures at launch time properly. 1986. So, the parameters that you're talking about changing those design parameters, the way you would change the way that it's built? Or are they operational parameters like flow rates and cutoff times?

They tend to be a mixture of both. I mean, operating parameters are very easy to change by a system and so when I would put something on the test stand, you're immediately going to try different frequencies, different temperatures, different flow rates, and you collect a lot of data for that and learning for that changing the design is a little bit trickier because you have to make a change in between. And so when we sort of look at those problems, we basically recommend, you have to change the design if you want to see what happens if you change the design. But there's already a lot of information and data that you can get for the system when you just change the operating condition. So, it's a mixture of both.

Now, you've mentioned turbulence and Black Swan and we're talking about devices that have to have a high degree of reliability, because there can be safety of human life involved very easily and possible catastrophic events. So, reliability in the face of wide range of operating conditions has to be considered. And yet some of this physics as I understand it turbulence like Navier-Stokes has chaotic behavior, that is impossible to predict or hard to predict or are you making it easier to predict - essentially, what's your contribution towards reducing the possibility of some black swan event that would have serious consequences?

So, one of the fun things I've sort of done a while ago is I sort of joined the Stanford Turbulence Research Centre, where I spent three months with the sort of top brains in the world, just trying to understand why turbulence is so incredibly hard to understand. And the funny thing is that you very quickly realize there's so many super smart people trying to understand turbulence, and nobody really does. So, I essentially converted from the competition has two fears, to be the smartest turbulence researcher, I'm going to sort of play the joker card and go into, guys, I'm not understanding turbulence, I'm going to carry out experiments with a wind tunnel over there. I'm going to create a deep learning model learning from those experiments, and then the model has understood what's going on. Okay, fine, I haven't, but I have a model that has and so I can now use this to predict turbulence better than you guys can. It works surprisingly well. It's not popular among physicists, because it essentially is cheating in their opinion. But the interesting thing is, and I mentioned my Honeywell gas meter example before, is that people have tried to understand how the turbulence behaves in those pipe flows, and try to build in different ways to understand and to channel the flows to make it simpler. But the reality is, when you have a gas meter in somebody's home, and the piping in the wall, and the temperatures are all different, for every person that you look at, it's just remarkably complicated to understand exactly what the flow is going to do, and how it's going to behave. When you try to model that with simplified physical models, you just don't get it and the physical models that we have, because turbulence is so incredibly complicated, also doesn't quite cut it. And so what we do is we essentially say, look, there's millions of those things already installed in people's homes and we've got so many hours on the test and I'm going to build a data-driven or a machine learning model based on all of this information of what's actually happening, to predict and understand what's going to happen in different scenarios. And that reduces the risk of bad events, because I have a bigger pool to learn from, right, I'm not just taking simplified physics in a computer simulation to tell you what's going to happen, I'm going to look at reality of hundreds of actual situations to look at what's been happening there in order to help you improve your decision-making and as such, it makes engines more reliable. My favorite example here would be that Rolls Royce and the entire aircraft engine industry has been switching their modelling and maintenance to live data for years and the amount of aircraft accidents that we have per year has been plummeting, like from hundreds per year to near nothing. Because this real data modelling, while it does have the disadvantage that I as a human can really understand what the algorithms are up to does better model the system's complexity, its degradation over time, the various volatile influences that the different components have. So, the modelling gets better and as such, the risk of rare events goes down.

So this makes me think about what else is like this sort of complex physics where some of it obeys straightforward rules and some of it is chaotic and I think about like economics or maybe biology; and is there an application of your work in those fields? And is that something you're interested in?

So, before I sort of started on my PhD in doing machine learning for aerospace engines, I studied computational finance. So, my interest in those methods is coming from the quick summer internship at a hedge fund, where you realize, these tools are remarkably powerful, and they can model really complex systems. And so engineering is learning from finance, versus the other way around a little bit here. So, we've sort of taken the deep learning that they've already got in operations to self-trading algorithms to a degree of sophistication that is remarkable, and sort of taking it back into hang on engineering needs this sort of stuff, too. So, I think in finance, yes; biology I've seen I've mentioned Google is doing this like personally, as a mechanical engineer, it's not my forte to sort of look into biological applications, where my passion lies at the moment is with anything that is related to fluid dynamics problems, because as we know, the Navier-Stokes problems are one of the millennium problems that are remarkably complicated and they do affect a lot of things from car aerodynamics, to aircrafts to flow meters to gas meters, to ketchup bottles and soap dispensers and deodorants and inhaler spray so fluid movements is really like fluid and fluid like aerodynamic movement, fluid movements all around us and that is currently the area where I think machine learning can add a lot of value over current modelling techniques.

As, you said there, a lot of this work was started in the finance industry because they could smell the money involved and Wall Street tends to suck up a lot of the people with the ability in this. So, I was interested to hear you refer to your passion for this work here. Because that is something that's clearly driving you more than the way that it could be leveraged, say, for Goldman Sachs and maybe I could phrase this as, what sort of difference could we see in 10 years as a result of the unbridled application of the technology that you're developing here? How would that show up in our world?

What a beautiful question. So, let me think about for a moment. So, if I go through the applications that we've already worked on where I've seen potential, let's sort of go through various areas and industries that I've had the fortune to look into. For example, soap dispensers, were remarkably hard to understand what even such a tiny thing does in order to always give me the perfect amount of soap. I know, it's a tiny example, but surprisingly hard to model because people haven't had the time to think about how the physics in cosmetic products work. And one of the things you can do is help them train AI models that understand that physics and that can make most cosmetic products from deodorants to soap dispensers significantly more durable, more accurate, and so on. When I say more accurate, I was probably let's go into the next industry: medicine. One of the areas where we've just published a great case study is trying to understand how a drug is distributed in your lungs. Because it's very complex to model if you're trying to get medicine into a person, how the sort of airflow from an inhaler actually results into how much drugs have been delivered. Very complex, very difficult fluid dynamics problem that actually can improve lives can help save lives and can improve like the delivery of medicines. And that's one of the case studies we've just published, where you can see that that works a lot better and you can understand that better. Apart from medicine and consumer products let's go into automotive, where lower drag, like less consumption, less fuel consumption, the better understanding of combustion works in the engine, how long batteries last how to optimize battery performance, obviously have impacts and margins. We've had this by somebody used to get 1% less emissions out of an entire car range. It's not a lot, but combustion engines have been around for a long time. So, it is a margin and if you sort of ramp it up, it becomes more and more, we've looked at aircraft turbine blades, where if you can better understand how a turbine blade degrades over time, using these kinds of test status, it's been estimated that can get up to 2% more out of them, which very quickly results to 14 15 million tons of CO_2 per year. So, I think in automotive and aerospace, it's sort of solving those intractable, those very complicated physics problems where I quite don't understand the physics yet allows me to build better products and I think that, again, is the Honeywell thing that I mentioned right in the beginning of a gas meter that was previously inaccurate, because it didn't measure how much gas people in the US in their homes measure as good as it was supposed to be and they could finally build a more accurate product with this new technology, which is more reliable, by understanding and modelling the physics. And so from getting more accurate consumer products in your home that measures your gas consumption in a more accurate way to better aircraft to better automotive, we are already pretty good in a lot of scientific areas. But this can always - it's the little extra, it's the sort of little chunk at the top of the scientific discovery that we currently can't tackle and that gives you improvement, sometimes marginal, sometimes radical improvements over the sort of state of the art that we currently see.

Basically it sounds like mastering physics with artificial intelligence, and machine learning and I'm excited about the potential that has for bringing us safer, cheaper and more effective things to market and even improving carbon footprints. It's been a fascinating discussion. Richard, where should people go to find out more about what you're doing?

My company, Monolith has a website, monolithai.com. You can also look at my personal LinkedIn profile where you can find access to my thesis projects that I've worked on in rocket science. Or just yeah, reach out to me directly.

Absolutely. Well, I thank you very much for coming on the show. That has been a terrific discussion and good luck with the next adventure in whatever rocket launch you get to be involved in next. Thank you, Peter.

That's the end of the interview. It hardly seems these days as though it would be controversial to use machine learning for something as critical as the design of parts used in machines that carry people or could kill a lot of people if they go wrong, but it's worth remembering that deep learning has only been around for a decade, roughly, and before that was so novel that the idea of trusting it for something this important would have been a hard sell. I think of it as another demonstration of how AI is maturing and fleshing out its markets, and the incredible diversity of those markets just speaks to the incredible versatility of AI.

In today's news ripped from the headlines about AI, artificial intelligence can predict heart problems from your voice. The American College of Cardiology reported that researchers from the Mayo Clinic trained the Vocalis Health algorithm to analyze more than 80 features of voice recordings, such as frequency, amplitude, pitch and cadence, based on a training set of over 10,000 voice samples collected in Israel. In previous studies, researchers identified six features that were highly correlated with CAD. For the new study, researchers combined these features into a single score, expressed as a number between -1 and 1 for each individual. One-third of patients were categorized as having a high score and two-thirds had a low score. They found that people with a high score were 2.6 times more likely to suffer major problems associated with coronary artery disease (CAD), a buildup of plaque in the heart's arteries, and three times more likely to show evidence of plaque buildup in medical tests compared with those who had a low score.

"We can't hear these particular features ourselves," said the lead researcher, Jaskanwal Deep Singh Sara. "This technology is using machine learning to quantify something that isn't easily quantifiable for us using our human brains and our human ears."

I'd like to tell you that I gave it a whirl, but it's not generally available yet. This is a great example of using AI to find patterns that humans can't, and it's obviously really valuable.

Next week, my guest will be Tom White, who is an artist studying the Algorithmic Gaze: how AI sees the world, and he can fool AI in some really interesting ways. That's next week on *AI and You*.

Until then, remember: no matter how much computers learn how to do, it's how we come together as *humans* that matters.

http://aiandyou.net