Welcome to FYI, the four-year innovation podcast. This show offers an intellectual discussion on technologically enabled disruption because investing in innovation starts with understanding it. To learn more, visit arc-invest.com.
Arc Invest is a registered investment advisor focused on investing in disruptive innovation. This podcast is for informational purposes only and should not be relied upon as a basis for investment decisions. It does not constitute either explicitly or implicitly any provision of services or products by arc. All statements may regarding companies or securities are strictly beliefs and points of view held by arc or podcast guests and are not endorsements or recommendations by arc to buy, sell, or hold any security. Clients of arc investment management may maintain positions in the securities discussed in this podcast.
All right. We're joined today by Naveen, who's the founder of Mosaic ML.
Naveen, do you want to just start by introducing yourself and your background?
Sure. Yeah. My name is Naveen Rao. I'm the CEO and co-founder of Mosaic ML.
A little bit about me. I think I'm maybe one of these crazy people who was around a little bit before AI kind of took off and was always fascinated with this idea of building synthetic intelligence. As an undergrad, I was electrical engineering, computer science, but I even worked on neuromorphic computing back in the 90s as an undergrad. I came out to the Bay Area,
worked in startups and built a number of different chips and software stacks and really kind of, I guess, the basis of a lot of technology we have today was being built at that time in the early 2000s, late 90s. After about 10 years in the industry, I decided I wanted to go back and understand more about this idea of synthetic intelligence or artificial intelligence. I quit my job with a kid and a half and went back to grad school to get a PhD in computational neuroscience. Honestly, it was an amazing experience. I really loved grad school. I can't say everyone says the same thing, but I really loved it. I think that was really, gave me a good basis to kind of start putting computer architecture, software, engineering capabilities together with what we understand about the brain to kind of formulate what computation might look like in the future, what AI, how it would start to take shape. That was really what led me to start my previous company, Nirvana Systems, which was the first AI chip company. It was really this desire to build the right computational substrate to subserve intelligence. We're continuing the journey now with Mosaic. I'm excited to talk a little bit more about that.
And Nirvana, you sold to Intel, correct?
Correct. Yeah, we sold to Intel in 2016, became the basis of Intel AI, which we grew into a corporate brand, a new corporate division, all that kind of stuff. In retrospect, we probably could have gone longer and had even bigger success, but it is what it is. We made the best decision at the time, but it was an incredibly interesting experience. I think I learned a lot more about people than I learned about technology. Yeah, for sure. And why'd you start Mosaic?
Yeah, it's really continuing along the same theme of to really make artificial intelligence happen is you have to make it easy to use and digestible by many different types of industries. Something that's in the research world that's an esoteric topic is never going to be, is never really going to change the world. You have to do that through products.
And one fundamental piece was the usability and difficulty of achieving this state of the art was very inaccessible to most companies. And these aren't dumb companies. They're companies

have tons of data. They have data scientists. They have people who are smart. It's just they don't know all the details of how to make systems work efficiently, how to scale across multiple computing elements, how to get the most out of every computing element. That's just not what they do every day of the week. And so by packaging those capabilities up, making it sort of straightforward to use that and get high performance out of it, you actually make a really strong economic argument. It's like every dollar you spend on compute becomes usable, becomes useful, actually moves your bottom line. And really, that was the basic thesis behind Mosaic. So it's a continuation of this theme of providing the right tools and substrates on which to build intelligence. For a customer that starts using Mosaic today, what did their life look like before Mosaic versus what it looks like with Mosaic? That's changed over the last few years. It's really about cobbling together a bunch of pieces of open source. So you can go get a cloud platform, AWS, Azure, what have you, get some GPUs. Okay, so now you got the GPUs. They have Kubernetes. Kubernetes works. It's kind of got a lot of rough edges. You have to do a fair bit of configuration to make networking work, to make sure you have all the right libraries loaded. So that takes a few months, actually. It's not simple. Then you go and download your favorite framework called PyTorch. Then you go get a hugging face model. Then you try to get those things to work. They sort of break. They don't scale. You try some different scaling libraries. They break. So this whole process takes years of experience to really know how to make it work. Like the companies that have been successful at making this stuff work have done it multiple times at Google or whatever. And so the talent to actually make this stuff happen was way oversubscribed. There's just not many people. So we want to make that whole process a lot easier by packaging this stuff up. And there's no good reason that it needs to be that complicated. It's just the level of maturity of the tools wasn't there. That makes a lot of sense. And obviously there's a lot of exciting progress in AI today. If you look back at the last year at 2022 and you had to pick one thing that was the most exciting breakthrough, what would it be? I know a lot of people are going to say chat GPT for this, but I actually don't think that was the biggest breakthrough. I think the biggest one was probably more the fusion models and being able to do kind of fusion of text and imagery. I think that's got a lot more to come, to be honest. Not that LLMs, large language models don't, but I think we're now in this refinement phase where we're dealing with the real world challenges. I mean, this reinforcement learning with human feedback is something that chat GPT really did. I'd say the model architecture itself really isn't that innovative. It's how they fine-tuned it and made it work, which is more where the innovation was. And really, that's a real world constraint. How do you make it give you better answers? How do you steer it away from kind of dangerous or bias topics, this kind of thing? And so that's really a real world set of technologies to make this stuff happen. Still got a lot of work to do that. So with that, I think maybe we'll jump into a conversation around chat GPT and these really large foundation models versus some of the smaller models that seem to be outperforming in certain domains. And then just talk more about kind of, yeah, in general, then we'll circle back to Mosaic and your experience and lessons learned and what you guys are building. So obviously, chat GPT just came out recently and it's broken headlines everywhere. Was that really a fundamental step function change or step function improvement
in the underlying technology or is this more of an incremental sort of progress in LLMs?

Well, it's interesting because I think apparently from the user, it looks like a big step function because now all of a sudden it does something pretty well that it couldn't do before. But really, that is a culmination of many small improvements. And so I think one of the big ones was reinforcement learning with human feedback, proper fine-tuning regimes to make these things answer more accurately and give you better responses that you might expect. And I think that was the big innovation with this. I mean, architecturally, it's really not that different from the original GPT-3, which really wasn't that different from GPT-2. It's just more scale. So the biggest innovations were really around that. And I think, you know, not to minimize that at all because that's what makes this work in the real world. You need to do these kinds of things to make these models actually usable. Yeah, there's rumors now that Microsoft is in talks to invest up to $10 billion into open AI. And presumably, a big chunk of that will be applied to compute as they try to scale up the size of the models even further. You know, when you go from from GPT-2 to GPT-3, which was a order of magnitude larger in terms of number of parameters, what are the challenges that you face as an organization that's trying to train these really large models? You know, I assume it's not as simple as like going to Zoran saying, hey, here's more data, train a bigger model. What are the complexities that the companies face and they actually get to do that? Well, okay, so let's talk about specifically that GPT-2 to GPT-3 transition. I mean, you know, when you're the first one doing it, like, you know, all the big labs, like open AI, deep, deep mind lab and Google brain, all of the unfair, you know, they had to invent a lot of the stuff as they went, because it just didn't exist. There were no good parallelization frameworks or weren't good, you know, orchestration tools, they had to kind of write a lot of this stuff. So that makes it really hard. Once you have some of that down, it actually isn't so bad, there becomes repeatable patterns of scale. So between GPT-2 to GPT-3, I think that's a lot of what they did was build that repeatable pattern of scale. Now, what are the challenges? So the challenges are that now your data sets start becoming a lot bigger. Now, in terms of just sheer gigabytes, they're not that big. We're talking terabytes of data, which isn't really that big by today's standards. But when you start thinking about what's in that data, you know, we're talking hundreds of billions of tokens. A token is, call it approximately a third of a word. A hundred billion words is, or a hundred billion, yeah, a hundred billion words is, you know, a hundred million books, something like that, right? So this is not a small amount of knowledge. This is basically pulling together that amount of data, cleaning it, making it into the right format is actually not trivial anymore, because it just comes from so many different sources. What's really interesting is that by some accounts, we've actually hit almost a limit of what's scrapable, meaning that if I go on the internet, I go on Google and I just try to pull every possible tech source I can, I'm running out, right? The largest models, the largest and chill models were trained with over a trillion tokens. So we've literally hit the limits of everything that's on the internet. I never thought that would happen, honestly. I thought that would just be like an infinitely scaling universe, but it's actually, we've hit that limit. So what's interesting now is that with the size of models that we can build, these 175 billion parameter models or 500 billion parameter models, we actually have the capacity to represent the internet's worth of data. That's the scale we have here. So how fast is growth going to be? I don't know. It's kind of an open question now. It's interesting looking at the chinchilla scaling paper. It seems like there's, you know, in a lot of the models that have been trained that

optimized for large number of parameters, there's probably, you could argue, they're under-trained in terms of data or under-supplied in terms of data. So if you're starting to run out of data and you want to make the models, you know, larger and more performant, like what do you
do? Do you try to create synthetic data or how do you approach that problem?
Yeah, that's a great question. It's an open topic. So I'll tell you a little bit about what we did with PubMed GPT. So this was a project we took on with Stanford Center for Research on Foundational
Models. Basically, the task is to take the MedQA evaluation, which is basically the U.S. Medical Licensing Exam. It's what human doctors have to take to be called an MD. And we trained it on PubMed data, which is basically all the primary literature papers from 1970 up to, I think, 2021. And this amounted to about 50 billion tokens of data. So we trained a 2.7 billion parameter model, which is much smaller than GPT-3. But the way we were able to get more performance was actually doing multiple passes on the data. And, you know, you can do some modifications to ordering and things like that. But it actually turns out you can squeeze a lot more out of the 2.7 billion parameter model by doing multiple passes. And so that's one strategy. Synthetic data is an interesting concept. I mean, the hard part is that you're trying to expose the model to more parts of the distribution, right? All the nooks and crannies of the data are what you're really trying to find. When you build synthetic data, you're kind of recapitulating the statistics of the data on which it's modeled. So you're not really creating new nooks and crannies necessarily, right? That's not always true when you're using simulation data and things like that. But many of these sort of resampling techniques don't give you new parts of the distribution. So I'm not sure they actually improve model performance all that much. Again, still an open topic. And there are lots of people coming up with cool techniques around this. So I think it is an area to explore for sure. Yeah, I think one of the other areas that's interesting is audio, right? There's a lot of audio recording on TV, on podcasts, like we're recording now on YouTube videos that you could translate into text and possibly use additional training data. So it'll be interesting to see where that goes. On the sort of model size topic, do you think we're, it seems like there's this sort of bifurcation in that there are really large models like what will be GPT-4 that are pretty good at like a broad range of tasks versus smaller models that are specialized? How do you think this landscape is going to evolve? Do you think we're going to end up with sort of like large, really large foundation models for just kind of general writing assistant tasks? And then specialized models to help us answer medical questions? Or do you think there will be some tilt in either direction? So I mean, I'm personally pretty biased on this. I don't see a world where there's one large model owned by one company. It's just not going to happen. Because there's no way one company can imbue the expertise of every domain into one model. Just not possible. I envision a world where there's millions of models built on millions of data sets that are experts at many, many different tasks. And I think that's actually, there's a lot of advantages to that. It's kind of, you can look at nature and that's kind of how it tends to work. These decentralized, many shots on goal kind of thing, not like one big shot, right? Typically. So there's a few reasons why I believe this. So one is, we talk to customers all the time that are in the enterprise space. Data provenance is super important. It's well known that large language models can regurgitate aspects of their training data. This is actually a

feature. It's not a bug. This is something we want. They can memorize chunks and bring those chunks back out when they're prompted the right way. It's not that different from a human, right? I mean, when a human goes and reads a book, they go and cite passages, right? So this is a good thing,
but what that means is that if you're in a banking context or an intelligence,
you know, US intelligence community, you can't use a model that was trained on Reddit.
It can have all kinds of junk in there that could regurgitate when you don't want it to.
And so you need to have some control over the data provenance. And so you want to create a model that's safe, that's clean for these different purposes. And I think that's really important
for things like medical applications, right? I don't want something that was trained on,
you know, some misinformation that came from the internet. It might be good in terms of syntax of language, but not good in terms of content. So this kind of becomes very important.
Another aspect is the way every company has to work in their specific field is that they have to be able to compete with the other players there. They compete by maybe gathering new kinds of data by driving different kinds of experiences to their users or their customers. So there's a differentiation that has to happen here. When they build their own data sets, they actually want to bring out unique insights that their competitors potentially don't have. So using the same model from everybody else doesn't get you that. You're kind of building on the same foundation. You're never going to have something as that different. You can fine tune and you can do some stuff around that, but you can only change the model so much. And I would say the last one, which is much more of a practical consideration, is that performance and speed of smaller models is just far superior. So, you know, Codex, the code completion neural network that's behind the GitHub code completion tool is about a 12 billion parameter model. Could you go bigger? Sure. You probably get better performance, but the problem is then it becomes too slow. It doesn't give you the outputs while you're typing. So I think this is why it makes sense to actually build smaller models that are potentially much more performance. And so I think we're going to see this whole universe of there will be some large models that do a lot of different things, and they will be useful for some set of tasks. But I actually think the long, much longer tail is going to be served by smaller models built on data sets that are clean, curated, and become experts at a particular domain. Seems like the other consideration is just margins, right? If you have to run inference through open AI every time you want to respond to a customer, you know, that can get expensive. And I saw some estimates that chat GPT costs about, you know, $1 per per query, or one cent per query response and inference, which one set doesn't sound like a lot. But, you know, when you get to scale, like that can be cost prohibitive and expensive.
And so it seems like that's that's another consideration as well that tilts towards
both smaller models and also owning your own model. Yeah, absolutely. I mean,
this is something that we're working on at Mosaic is, you know, building inference pipelines beyond the training that are basically, you know, charged on a compute basis,
on a compute our basis, not on a API call basis. And I think that idea made a lot of sense, especially when it was very difficult to build your own models and have anything that's even close in performance. But as that gap closes, it makes more sense to potentially own it for all the reason we just talked about. Yeah, that makes a lot of sense. And in terms of data and data provenance, you know, there's been a lot of question around like modes and AI. And it seems like

data is one of the modes that does exist. You know, if you have your own proprietary data set she'd use to train a model, and then you have sort of a data flywheel that you can use to, you know, continually curate additional training data, seems like that that would give you a mode. Is that something you you are seeing as well? Yeah, I think that is true. I think data is a part of the mode. I would as way I would put it, it is, it's kind of a necessary but not sufficient part of the mode. So once you have the data, like, let's say, let's say I was given a bunch of like genomic data. I mean, can I do something with that? Probably. Am I going to be as good at understanding how to fine tune that model as someone who really understands genomics? Probably not. And I think there's that's the other aspect is how you how you wield that data, right? It's like, first, you got to have the sword to fight. But there is skill in how you wield that sword. And it's the same kind of thing here. I think that's really going to be the mode on how people get really, really good at making their data usable in the form of an LLM. Building the right LLM that solves the problems that their customers care about. And looking at the customer base that you guys have today, how do you see customers actually using LLMs in practice? Yeah, this is an open question too. It's something we're learning every day. I mean, I never underestimate the creativity of the world. When it comes to this kind of stuff, like you give people this this basic, you know, building block, and they can they come up with all kinds of cool stuff. I mean, there's there's things like personalized, you know, avatars that people came up with. Obviously, that blew up. We have a customer in that space doing doing kind of interesting things. Then of course, there's personalization of even your corporate of text, we have a company that's doing some of that stuff. So like, let's say I'm an influencer, or I write a lot of blogs or write books, you can actually bake a lot of that into a book and you can almost have a conversation with the author through that LLM, which is kind of neat, right? I think, you know, code completion and better tools for writing, you know, more testable, less buggy code is a huge one, right? Kind of bringing some experience from the entire community expressed through the LLM to the user who's writing a piece of code. It's like, don't write it this way, write it that way. That's a really interesting use case. We're seeing things like credit risk modeling in financial sectors. You know, I think anything where you interact with text, which is basically anything can can be upended and improve with these kind of capabilities. So I'm actually seeing, honestly, faster uptake in even old industries than I expected.

And this is more of a, I guess, a macro question in some sense, but, you know, it seems like we're approaching a point, at least in certain domains where AI is able to, you know, augment a human, make them five, 10 times more productive. If you take software engineers, for example, you know, we saw co-pilots doubling the productivity of software engineers, at least within coding tasks. And so if you extrapolate this out, right, if you have, you know, a team of 10 engineers and all of a sudden you can get the same output with five engineers, what happens? And it seems like there are kind of two directions. One is that we see a reduction in knowledge workers because you need fewer knowledge workers to get the same output.

The other is that we see companies sort of harness this technology and continue maintaining the same workforce, but, you know, increase the productivity of that workforce and build more products. And, you know, I don't think anybody has exactly the right answer. I think if we look at history, it would suggest that, you know, technology tends to improve wages and improve

standard of living. And it's probably a net positive for everyone. But curious, where you sit sort of on this, on this spectrum of optimism, do you think that, you know, AI will completely, you know, disrupt the economy and make people unemployed? Or do you think it will sort of be embraced by it by humans and actually lead to a net increase in GDP?
I'm very much in a camp that elites the net increase in GDP. And of course, you know, again, I can, I can talk about my bias in this field. I've been at it for a long time. But I honestly don't see a path that it does destroy the world in this like negative way where nobody know what has anything to do. I mean, every time there's been a technology transition, right? If you look at articles around the time of the industrial revolution, you saw a very similar rhetoric. Actually, it was like, you know, the guy who was plowing a field no longer has to do that. And then one guy can drive a tractor. It's like, yeah, but you don't have people doing that. It changes and shifts what what people do. And there are some growing pains in that, I think, you know, some workers who have a knowledge base and that gets disrupted very quickly, they have to retrain or or not. And that causes some pain. But overall, I think it increases the amount we do as a species, which allows us to support growth as a species allows us to support new new endeavors like moving out outside of our planet, building new technologies that, you know, can support this growth, like fusion, which just happened, right? The big breakthrough that just happened. So I think all of these kind of things go together, we build better AI tools that allows allows us to build new new things like fusion or better space exploration. So all of this kind of goes together. And I see it as like, it's like a manifest destiny of what our species can do. The other way to look at this as well is, you know, we really need it just to maintain current output. If you look at population decline, like that's one of the more existential risks that humanity is facing, especially in modern developed countries is population decline. And I mean, if you look within certain domains, like look within healthcare, there is a massive shortage of doctors. And so you really actually need AI just to maintain the current output with a declining workforce. The other thing that's interesting is, you know, AI is in many ways like this great equalizer, and that, you know, something that used to require like a lot of education and, you know, extensive domain knowledge, you know, can now be accomplished or a skill can be acquired by somebody that has, you know, internet connection and motivation, right? We saw that, you know, historically, like with the internet, right, it democratizes access to information. And it seems like AI in many ways democratizes access to, you know, unique skills. You know, you no longer have to be an artist to generate art. And I think this will persist across many domains. And so it's sort of the, I guess the optimistic way to look at that is like it unleashes entrepreneurship. If you have the motivation to create something, you no longer need the skills, right? You can use AI to augment the skills that you do have and produce something useful. Yeah, absolutely. And I think the danger of some of this is that you have centralization of resources. And this is why I philosophically very much am happy about what we're doing at Mosaic is that we are building tools to enable more people to have access to these technologies. When it's all centralized in one or two or three players, that creates a huge power dynamic. And, you know, that actually happened with the internet
too, right? I mean, yes, at one, in one level, it democratized access. But on the other level, it actually centralized a lot of things, which caused some problems, right? I mean,

we're seeing the aftermath of that and politics and all kinds of disinformation and this and that it allows centralized points to have influence undo influence on the population. AI is yet another lever on top of that. And so figuring out how to make people be able to wield these technologies themselves and and do so at various scales, small scale, all the way to large scale is how we kind of mitigate that problem. Yeah, completely agree. And I guess just shifting shifting a little bit. Like one of the things that's interesting about AI is historically, at least in recent times, like software hasn't been compute constrained, right? Like we haven't really like if you want to build a SAS application without machine learning, like you don't really run up on the limits of compute capacity or compute capability. AI is running up against the limits, or at least it appears to be in its cost prohibitive, its time prohibitive to train some of these really large models. The good news is there are companies like Mosaic that are making training more efficient. And just curious, like, how do you actually do that? How do you take something that, you know, would use X compute and make it use a fraction of X compute, but get the same same performance? Yeah, so actually, before I go into that, I disagree that software hasn't been compute constrained. I think, I think Moore's law and software have kind of co evolved to line up in terms of timing. But if you run a piece of software on 10 year old hardware, it doesn't work so well. It is actually constrained by that. It's just that we were able to meet this. That's very true. So I think, you know, we had a very stable environment for a long time, and now that's been destabilized. The curve just kicked up even faster, right? And so we're sorting it out. And to your question on how we do it at Mosaic, it's actually interesting that when we started the company, we were looking for sort of a silver bullet. It's like, you know, one of the folks that was around from the very beginning, and is now the chief scientist of the company is Jonathan Frankel. He published a paper called the lottery ticket hypothesis, which was a seminal paper in the space of compute efficient learning. And we were kind of looking for that one or two set of methods that would be like, Oh, my God, this is so much bigger, so much better. Reality is it comes from many different small tweaks. Some of them are are sort of basic hygiene tweaks. You know, you know, getting the right packages, configuring the system the right way. Some of them are understanding patterns that are seen over and over again, and actually optimizing those. So very low level software. This is this is nothing groundbreaking. This is standard stuff that people have done. We package it up and make it very usable. Then the next level of stuff is where we actually start to modify the math. So there's sort of, we separated the world early on into math preserving and non math preserving and then even pseudo math preserving. And so pseudo
math preserving is like going from 32 bit numerics to 16 bits for a lot of your calculations. It's sort of math preserving, but actually not really because when you change the precision, you actually change the math a little bit, but it's an acceptable kind of change, right? And then there's stuff where we're even talking about modifying, you know, the back propagation algorithm pretty significantly. So there's a whole set of methods. And so like the way we achieve this with some of our computer vision methods, I think we're an order of magnitude faster, more compute efficient than
the standard implementations. We did that through about 25 different methods, some of which are, you know, changing things like the convolution size, making anti aliasing data before goes into a convolution that makes the convolutional learning more efficient. Some of them come from

how we present data to the network. And actually, I think this is going to be the big theme for the next year or two or three is neural networks are like a brain, right? When you if you present a child with with too complex of a data, it's the child's not going to learn anything. You have to present the simple stuff first and then present the hard stuff later and build upon it. And neural networks no different from that. You got if you present too complicated of a story to it early on, it's just not going to learn anything. So actually, stepping this stuff up, like we do things called sequence, like sequence length warmup, where we actually present smaller sequences and learn those kind of primitives before we start presenting bigger stuff. So combinations of things like this make a big difference. There's actually some really interesting things that the brain does that we don't really do in neural networks today. So one example is, if I were to teach you to play tennis, assuming you don't play tennis, and you never serve the ball, you know, you probably
know how to throw a ball. You've done that before. So you're not going to practice that. But you are going to practice the whole motion and going behind your back and trying to swing. Because maybe that's
not something you've done. Our brain is really good at saying, Okay, I don't need to spend time and attention and energy on the things that already weren't and only spend that time and attention energy on the things I need to learn. And we don't do that in our networks today. We kind of throw a bunch of data at it. It all turns through regardless of whether that data point is useful to learning or not. And so I think this is going to be a big theme going forward is,
can I decide, is this data point useful for learning? If it is, then I'm going to use it,
and I'm going to learn as much as I can from it. If it's not useful, don't waste time on it.
So these are the kind of themes that we've gone through in making things more efficient.
And when you have conversations with customers who want to train their own models on your platform,
and you sort of talk about the efficiency improvements, do you see customers optimizing
for performance or for cost? It's all over the place. We're trying to form hypotheses around this, which would be more important? How do we pitch it? And it really depends on the customer. Some customers care about the performance of the model at all costs. And sometimes they care about
the performance of the model within a budget of parameters, like co-pilot or the PubMed GPT.
We wanted a small model that had performance. And there's a lot of reasons for that because of the downstream constraints of latency. Sometimes, I mean, there are certainly applications where it's like, cost be damned, I just want the best thing. I would say that's kind of been the open AI approach. Spend a billion dollars, prove that it works.
Most enterprises are somewhere in the middle. There's some mix of these things, and it's not the same for each one. And so I think the way we now position things is we give you the tools to make these trade-offs. You can say, this is going to cost 5x more, but you're going to get 30% better performance. So is that worth it to you or not? And I think this is the kind of abstraction that enterprise customers seem to want. They want the ability to say, all right, well, in this particular application, I really need to make the performance hit this certain level. We've seen that. We had a customer who had to get out to a particular threshold,
or they couldn't deploy something. Okay, so spend what you got to spend to get there.

In other cases, I just want to see if this thing is going to work, or there's nothing.
So I want to build something that shows that I can make an application here. Then it matters how much you spend. You don't want to spend too much. It makes sense. And it's different for every use case, I imagine. If we look at cost declines historically, between, I think it was 2012 and 2020, your training cost declined at about 65% annually. Do you think that's going to continue through 2030, or do you think that rate of cost decline is going to change?
I think it's going to accelerate. We've had discussions as we've gone through this story, and I think you guys are undercalling it personally. So to break it down a little bit, Moore's Law, when it was chugging along 2x every two years, rather, is about 40% per year. So arguably, we're less than that now, so I'll call it 30%. I would say architectural advances have given a lot of the year-on-year improvement. So going to a two-dimensional architecture, a tensor core architecture, that's what we did in Nirvana, was a big step function. So when you smooth that out over the last few years, you end up with another 10%, 15%, 20%. Software improvements,
fusions, that kind of stuff gives you a little bit more. So that's 60% kind of makes sense.
When we started bringing on these algorithmic stacks, we saw orders of magnitude. Within a couple of months, we were able to hit 10x. And what's really interesting is ML Perf, it's an industry benchmark. We actually won it in computer vision with ResNet. We actually did it again with BERT just recently. The BERT submission, we decided to do literally two weeks before the deadline, because we weren't sure we want to put the time in. We're like, all right, we'll just throw some stuff together that we did months ago, and we were of state-of-the-art. So I think that shows you that there is now a new well to pump, right, that you can basically say, okay, I can put these pieces together in a new way and make things more efficient. And so I think we're going to be in this regime of three to four x improvement per year. So that's, I don't know how you want to say that in terms of percentages, depends how you count it, but it's more than 60% per year.
A lot, a high percentage. Yeah, it's interesting too, like if you look at throughout sort of the history of the digital economy, it's like Moore's Law was such an important driver of growth. And this is true across, I mean, many domains, right? Like if you look at the adoption of electric vehicles, like you can create a, you look at a cost decline curve, and it's pretty well tied to growth of that industry. And that's been true for software as Moore's Law has declined. And now we're
seeing, you know, double triple the rate of cost decline with training that we saw of just Moore's Law alone. And so it's like, it seems like this will sort of manifest in two ways. One is, you know, the very, very large models will continue to be, you know, push the boundaries of performance,
but then it just becomes cheaper to train models, right? And the, like, you know, if you look at, even now it costs you a few hundred grand to train a model, like, it's not a lot, but like, it's significant. But if it costs $500, like, how many models are you going to train? How often are you going to retrain your models? It's going to be a really, really crazy five years as I think more and more companies look at AI and look to sort of embrace them in different ways.
Yeah, 100%. And I think you're making our case, right? That's what I mean about millions of models, right? You make this easy, you make it cheap. It's just going to blow up. It'll be applied to everything. You can personalize everything. Yeah, it's like, you know, the great, the history of,

of the password is actually quite interesting on computers. It went back to, like, when computers were really expensive to run. And you had, you know, maybe one computer at a university, and you had an allotment of time. And so you had a password that you were assigned to use that computer. And that's actually the first, I think the first reported hacking of passwords was back in, I think it might have been MIT or one of the research universities where some computer scientists wanted more time on the computer. And it's, we're sort of at that moment with AI, right? It's like, it's still somewhat cost prohibitive and hard to train these models. And like, you have to be a computer scientist to use the computer at your university. And same thing here, you have to know what you're doing to train these large models, and you have to have access to a lot of capital and compute capacity. But that's changing very quickly. And like, I mean, with what you guys are doing, like it's becoming a lot less complex and a lot cheaper. And so I think that the way that, you know, computers sell mass adoption because they got cheaper and they were easier to use, probably see the same thing with AI. Absolutely. Yeah. And I think, you know, the algorithms are an important part of that, how we actually make these computing elements, you know, learn from data, right? You know, what's really interesting is, so I am a neuroscientist, and, you know, if you look at a rat's cortex under a microscope, and you look at humans, it's really not that different. The cells, the structure, the layers, they're all the same. We diverged from rats, I think, I don't know, 120 million years ago on the sort of evolutionary scale, maybe more than that. And the fact that the cells and the structure has been relatively stable says something very important. It wasn't the underlying compute element that actually gave rise to human intelligence. It was the organization of those compute elements. And I think that's what we're seeing now is that Moore's law is slowing, right? We're not building transistors that have a huge economic advantage on the next generation anymore. They're a little bit denser, a little bit more power efficient, but that comes at a cost. So the economics aren't playing out anymore on the substrate. So we sort of halted that. But now it's really how do we put those things together? And I think we're just now scratching the surface, because we have to. Scaling of devices was a really quick, not quite easy, but sort of a straightforward way to keep making economics chug along. And now we've got to think a lot more creatively. And I think computers
in 30 years are going to look significantly different from what they do today. I don't think they're just going to be numeric manipulation machines. They're going to be something very different. Do you think, just touching on this neural architecture for a second, there's a lot of debate on Twitter on whether AI will become sentient, however you define that at some point. What's your position on that debate? I mean, first, sentience or consciousness is such an ill-defined term. I don't even know really how to approach this. I mean, it's funny because as a neuroscientist, every first year neuroscience grad student comes in and wants to ask that question. And then it's basically beat out of you. It's like, don't ask that question. It has no real answer. But that being said, I think there is, there are attempts, right? Like Kristoff Koch and others have tried to codify what this might really mean. And there's definitely something about being attached to the real world, like some kind of input representation, action space and this loop that has to be closed for something to be sentient. So, everyone's like, there's a lot of people saying, oh, GPT three or four or whatever is going to be sentient. I completely don't buy it at all because it's static. It doesn't learn while you're using

it. I mean, there is a loop, right? You can take in data, you can then fine tune it, you can modify it, and you can put out different kinds of data, but it doesn't actually see the modifications into the real world. It doesn't have an action space really. And so there's still a ways to go there. And that's okay. It doesn't mean that we're failing. I think that's a great step on the way there. We're just not there yet. But I think there is this idea of embodiment also, where AI and computation needs to kind of act on the world, see the results of that action, and then represent that, come up with a new action. That to me feels like what a true conscious intelligence would need. It's a better defined framework than the debates I've seen on Twitter.
Well, yeah, that's not surprising. Again, democratization of your voice leads to some of that those viewpoints getting espoused. For sure. And as we wrap up here, just want to circle back a little bit to your experience as a founder. This is obviously your second AI company. You spent a lot of time at Intel. If we rewind back to Nirvana, what was the transition of selling to a big company like as a founder? Yeah, I mean, it's an interesting one. I've talked about this before, but it's an emotional one. The first thing that hit me when I talked to my company was I told everybody that we're going to do this acquisition was it's just sadness. I broke down in tears in front of my company without any warning because it was like, oh my God, what would I just do? And so I think there's that one aspect. Just as a founder, you feel this intense ownership of a company that you've seen from day one. Then beyond that, I think I fought the struggle of the innovators dilemma daily. And it's just amazing to me that that keeps happening. Everyone keeps making that same mistake over and over again, but it's just human nature. In big companies, there's sort of an established revenue stream. There are things to defend, and you will get people who will defend them to their death, and they will run every kind of political smear campaign they can against you. That's why I said I learned a lot about people. And it was like, look, I just want to build technology. That's what I care about. As a startup founder, you kind of have these pure intentions many times. But to actually make it work in a big organization is very, very hard.
So I learned very quickly why these kind of innovations don't happen in big organizations. Status quo is very much defended. So I think as a founder getting acquired, presumably by a large entity, you will hit that. I have not talked to a founder yet that's been acquired by a larger company that hasn't hit that at some point. I mean, some companies do it a little bit more hands off. Every large corporation is different. Google has been like a lot more hands off with many of their acquisitions, which is sometimes not a good thing. They don't actually incorporate the technologies. I actually am very proud of the technologies that we did incorporate into Intel. Like, we did a great job there. We did bring those alliances in and stuff like that. But there's a frustration level that you're just not used to dealing with when you're in a small company. Because it's like the analogy I like to use is when you're on the highway and everyone's going 50 miles an hour, if you go 50 miles an hour, your life's pretty good. You sort of sit there and you drive. If you want to do 90, you're constantly running into the back of someone. You're passing someone. You're pissing someone off. That's what has to happen if you want to go 90 when everyone's going 50. So I think as a startup, you're kind of on an empty highway. You're just driving. That's the mentality shift that you have to take is like, no, this is not a you just go as fast as you can kind of thing anymore. Now it's collaboration. It's bringing people under your tent. It's like this whole political game. And I think that's just really hard for a lot of founders.
That's a great analogy. And I mean, if you look at like the companies that are threatened by AI today,

you know, I mean, using an example with Google, you know, everybody on Twitter thinks the chat GPT will disrupt Google search. And this is a bit of an extreme example, which I think maybe a stretch too far. But, you know, when when you're in Google, and you're a product manager that's trying to release some AI product, you're facing a lot of, as you put it, cars going 50 miles an hour. Do you think that that AI will go to the advantage of the incumbents? Or do you think it will go to the advantage of startups that are trying to disrupt the incumbents? And that's a very broad question, right? Like, Google is probably going to be really hard to disrupt. But if you look at other domains of, you know, take like Microsoft Word and look at all the new writing assistants that are coming out that are trying to steal market share there, how do you think this is going to play out just in general at a very high level as companies face this innovator's dilemma? Yeah, that's a great question. I think we touched on a little bit earlier, like some of the moats, right? Data is a sufficient and not sufficient but necessary part of the moat. I think what startups do is tend to get higher quality people driving the agenda many times. And so what happens is you've got on one side, you've got this massive mode of data. The others on the startup side, you have this massive mode of talent. And, you know, the question is, can talent do better than the mode of data or not? And I honestly, I think it's going to depend on the specific domain. It's going to depend on how quickly some of those incumbent players move. I mean, I will give Microsoft a lot of credit under Satya. They've done really well in terms of being unafraid to disrupt some of their legacy businesses, right? That's what you got to do. It's hard. It's really hard because you're holding a degree down your neck and you've got quarterly, you know, earnings reports and this and that, like saying I'm going to spend a bunch of money on something that hasn't been proven out is hard. Now, in more conservative industries like insurance or even medicine, I think it's going to be dominated by the new players because the incumbents are just too conservative. And again, I think it's a cultural thing, right? Like, you know, in insurance, it is by nature a conservative thing, right? And so if someone comes out and comes to the much better risk modeling framework using AI and they can be much more adaptive and using these technologies,
they're going to beat the incumbents hands down, even with their data modes, right? So I think it comes down to the culture of the business. That's a great way to look at it. And just as we wrap up a final question for you, if you go back on sort of all the experiences that you've had through your career and you could go back to day one of founding your Vana, what advice would you have given yourself? Oh, man. Okay. I guess there's a few points. One is, you know, keep the team lean and mean from day one when your gut tells you to do something, taking an action like, you know, employees not working out or, you know, a partner is going to waste your time, listen to that. You got to trust your gut. That's probably the first one. I mean, of course, it takes some time to develop those gut feelings. But like a lot of times are right. And you're regretted not listening to them. And I would say, you know, don't be afraid to go big when you talk to investors, you know, that big vision is what matters. And we had that big vision at Nirvana. But, you know, we're just almost afraid to talk about it sometimes, like, oh, they're going to think we're crazy. It's like, no, no, we are crazy. And that's what you want. This world is all about like that craziness becoming reality. And I think don't be afraid, you know, I don't do something for a crazy sake, do it because it matters. But like, don't be afraid of it being crazy.
It's probably the biggest advice I've given myself. That's great advice. And thank you for

joining us. If people want to keep following you and the journey, they can find you on Twitter. And then where can they learn more about Mosaic? MosaicML.com. We have a pretty rich blog series
there that we try to highlight some of these topics and discuss them. And we have a Twitter handle as well at MosaicML. And on LinkedIn, you can follow us as well. So take a look. We're always
posting content there. One of our researchers posts a, you know, every couple of weeks synopsis of all the big papers in AIML. His name is Davis Blaylock. You can follow him on Twitter as well. And you can see those updates. It's a pretty cool sub-stack. That's great. Well, thank you. Absolutely. Thank you for having me. ARC believes that the information presented is accurate and was obtained from sources that ARC believes to be reliable. However, ARC does not guarantee the accuracy or completeness of any information. And such information may be subject to change without notice from ARC. Historical results are not indications of future results.
Certain of the statements contained in this podcast may be statements of future expectations and other forward-looking statements that are based on ARC's current views and assumptions and involve unknown risks and uncertainties that could cause actual results,
performance, or events that differ materially from those expressed or implied in such statements. Thank you.
you
you
you
you
you
you
you