[DAISY THE GREAT](https://www.facebook.com/daisythegreatband/) 00:00 [BUILT MY HOME ON HOLLOW GROUND](https://www.youtube.com/watch?v=z-1sC1lkmKw)

CRAIG: 00:07 Hi this is [Craig Smith](https://en.wikipedia.org/wiki/Craig_S._Smith) with [Eye on AI](https://www.eye-on.ai/podcast-archive), a podcast about artificial intelligence. This week, we’re doing the first of an occasional series of sponsored episodes with select companies whose technology we think promises to have a significant impact on the AI space. We’ve picked [Determined AI](https://determined.ai/), whose product is an [end-to-end platform](https://thenewstack.io/determined-ai-promises-end-to-end-orchestration-and-model-management/) that gives deep learning teams the tools they need to streamline their workflows. Determined’s platform brings the moving parts of model development under one roof, from collecting and keeping track of data, to building and training models, to sharing and collaborating with other people, to efficiently managing limited compute resources, and finally to deploying models in the real world.

[Automated Machine Learning](https://en.wikipedia.org/wiki/Automated_machine_learning) – algorithms that automate the writing of other machine-learning algorithms – are an increasingly important tool in this scarce talent environment. You want your expensive ML engineers doing the stuff that machines can’t. Determined’s platform incorporates AutoML functions such as [hyperparameter search](https://arxiv.org/pdf/1502.02127.pdf) to speed up the process.

The company’s founders, [Evan Sparks](https://etrain.github.io/) and [Ameet Talwalkar](https://www.cs.cmu.edu/~atalwalk/), talked about their journey and their vision of eventually democratizing AI. I hope you find the conversation as interesting as I did.

DAISY THE GREAT: BUILT MY HOME ON HOLLOW GROUND

CRAIG: 01:48 I'd like to start by hearing who you are, what your educational backgrounds are and how you guys got together. And then how you came about to form Determined AI. And then from there we can talk about some of the science behind it.

EVAN: Sure thing. Do you want to go ahead with the introduction first, Ameet, and then I can give my own background?

AMEET: 02:06 Yeah, absolutely. So, I'm Ameet Talwalkar. I wear two hats. I am a co-founder and chief scientist at Determined AI. I'm also an assistant professor in the [Machine Learning Department](https://www.ml.cmu.edu/) at CMU. My background to get to where I am today was sort of - not a straight arrow. I was an undergrad CS major in college, but computer science was not as hype-driven and as exciting, at least externally to students, as it is today. So undergrad CS major, but then I worked for four years doing various things, and eventually came back to computer science, stumbled into machine learning for my PhD at NYU. When I finished my PhD, I was at a bit of a crossroads in the sense that even starting then, and I finished my PhD in 2010, I felt that a fundamental issue in machine learning was this fact that there were a lot of great techniques, a lot of great methods, a lot of promise in the field and of what had been developed over the last 50 or more years in machine learning, and other related areas.

AMEET: 00:03:02 But it was really confined within the walls of academia. So, it was really hard for people who didn't have PhDs in these very specialized topics to actually make use of these methods. And again, this is pre-deep learning pre all the hype that we're seeing today. But still, something that I felt. I didn't want to be working on research topics that might be really interesting mathematically, but for which I didn't think that there was actually real practical motivation for solving those problems. The real motivation was just, I want to prove something very cool mathematically. I ended up doing a postdoc at Berkeley.

 At Berkeley, they were starting this new lab at the time called the [AMPLab](https://amplab.cs.berkeley.edu/about/), which was focused on the intersection of machine learning and systems, which fast forward nine years to where we are today, this is becoming a big field. We just started a [new conference](https://www.sysml.cc/) at the intersection of these two methods. But back in 2010, nobody was doing this. It was just Berkeley. But it was really interesting cause it was really focused on building systems and being more practical about what machine learning should be built into systems that more people could use outside of say, academia. So, we were engaging industry and things like that. So, for me from there, everything I've done since was very much inspired by working with, collaborating with both machine learning people, but also systems people. That's where I met Evan and we started collaborating a lot. Maybe I can let Evan take it over from here.

CRAIG: 04:17 Yeah, I’d like to hear, Evan, your background.

EVAN: Sure, absolutely. So, I'm Evan, Evan Sparks, I am a cofounder and CEO here at Determined AI. Prior to getting my PhD at Berkeley in the AMPLab, I started my career really in quantitative finance. After a few years of that having some time doing hands-on applied machine learning, I found myself much more interested in the technical aspects of what we were doing than maybe the financial problems we were solving. So, I went from asset-management-land to work at a startup in the NLP space called [Recorded Future](https://www.recordedfuture.com/) where I was an early employee.

EVAN: And there was building models where the dataset was the Web. What I learned there and, having been a practitioner building models and shipping product around models for a living, was that as soon as we got to the scale of data where it no longer fit in memory on my laptop, all of my tooling broke down really, really hard. And I felt there was a good opportunity to go to grad school and invent the future that I wanted to live in. And so this is about the time that I went to Berkeley and started collaborating with Ameet. And we started working on distributed systems for large scale machine learning in the context of [Apache Spark](https://en.wikipedia.org/wiki/Apache_Spark) at the AMPLab. And so, we and some of our collaborators at the AMPLab, were responsible for the design and initial implementation of the standard library for machine learning in Apache Spark called [MLlib](https://spark.apache.org/mllib/).

EVAN: 05:39 And I continued in my PhD, both working with Ameet, but also working with some other professors and students at Berkeley on methods for automating machine learning in a bunch of different ways. So, this is automated hyperparameter optimization but also things like how do we optimize and build end to end machine learning pipelines and scale out these applications in a big way. And I think the thing that happened in that period from 2012 to now was, one of the most remarkable things we saw happening out in the industry, was the rise of deep learning, or re-emergence I should say, of deep learning as a primary way that people want to be solving machine learning problems. And this led to completely new tool sets being put out there, things like [TensorFlow](https://www.tensorflow.org/) and [Pytorch](https://pytorch.org/) and so on.

EVAN: People leaning hard on [GPUs from Nvidia](https://docs.nvidia.com/) to accelerate their work which they weren't really using before. And this complete change in the landscape of software and tooling and even techniques that people were using meant a change in the workload and the workflow associated with large scale ML. We saw that, as a company, we had been studying problems related to this in academia for some time, but as a company what we really saw was an opportunity to go and provide access to tooling that would really democratize the way people are able to build these machine learning applications. If you go to a place like Google or one of the other big tech companies, they all have fantastic internal tooling for their developers to build robust, world class AI-powered experiences.

EVAN: 07:08 But if you go to the rest of the Fortune 500, there's nothing there. They're left piecing together things off the open source. And we felt as a company, our duty is really to build that for everyone else and offer that same level of experience. Because we really do think that AI and machine learning, deep learning, all of these things have tremendous power across lots of domains and every industry can benefit from better data driven decision making and applications that are powered by statistical models. And so, figuring out how to bring that revolution forward across the industry is really what we're focusing on.

MUSIC: INTERLUDE

CRAIG: 07:49 So, at AMPLab there were both systems people and machine learning people and that got you thinking about the scalability, is that right?

EVAN: I think for me it was really, the need for scalability was a consequence of what I saw in industry where I would try to run these machine learning algorithms on bigger datasets and nothing was working right. Like none of the tools, the open source tools that were out there, could handle the scale. And so, we felt, naturally, that we needed to develop these things. I don't know what's your take on that Ameet?

AMEET: I had been thinking during my PhD on actually large-scale divide and conquer methods for machine learning. So it was a bit inspired by the fact that I was doing research with collaborators at Google research who were a bit ahead in terms of building their own infrastructure and obviously collecting huge amounts of data. So, I had been thinking about methods for parallelizing and distributing computation such that we can scale things up, but I was very much doing it from the point of view of a pure machine learning researcher. So, I was writing all my code in [Matlab](https://www.mathworks.com/products/matlab.html) knowing very little about what the impact of these methods were when you actually tried to use them in a system outside of say, Matlab. Basically any real system.

CRAIG: 09:02 I had heard you give a good example, an anecdote of, you create this model in something like Matlab and then turn it over to an engineer and he says, ‘Hey this is going to take far too much power, too much memory or something, for the devices that we're trying to put it on.’

AMEET: My own example of doing this, the story we would tell in 2011, 2012 is that machine learning developers were stuck in a so-called Matlab cage, which is this idea that I was really good at Matlab - it's too bad that Matlab is not, people have moved on to more open source things for the most part, but that was probably for the best - but I was very proficient at Matlab and much less proficient in the other tooling that was out there. Maybe things like [Hadoop](https://hadoop.apache.org/) or [Mahout](https://mahout.apache.org/) at the time. And as a result, when I really wanted to have my new methods have impact, have people actually use them, the path that I would take to try to get adoption and to release code was to jump through a lot of hoops to release basically open source Matlab code that people could run in parallel on distributed settings. And it turns out that there is a way to do it, but it's absolutely the wrong way to be doing things. But, it's what I would do because to me that was easier than trying to work with more robust open source tooling that was out there for the systems community. The barrier was way too high for someone like me.

CRAIG: So then when did you guys form the company?

EVAN: 10:20 So the company was founded in 2017 and I should point out that our third co-founder [Neil Conway](https://www.linkedin.com/in/nconway/) joined us right at that time. Neil, for what it's worth, we talk about the intersection of machine learning and systems and the need to get people from both sorts of backgrounds together in a room to solve some of these really hard problems. Neil comes from very much a really hardcore distributed systems background. He's been a [committer](https://www.enterprisedb.com/blog/who-contributes-postgresql-development) on the [Postgres database](https://en.wikipedia.org/wiki/PostgreSQL), worked on [Apache Mesos](http://mesos.apache.org/) quite extensively. And we think that that mix of skills, both, having done applied machine learning, understanding the theory, and really understanding the systems component at the deepest level, is foundational to the DNA of what this company is about. And so, while Neil isn't joining us on the podcast, I think in a lot of ways that footprint and fingerprints on the product that we've built is absolutely crucial. So, we started in 2017 is the short answer to that question.

CRAIG: 11:20 And the idea, if there are CEOs or CTOs out there listening, or data scientists working at big corporations, is that there is a workflow in building a machine learning model and deploying it. And there's a lot of time spent building tools internally to make one thing talk to another or to capture audit trails and all of these different things. Whereas you guys provide that in a platform interface. Is that right?

EVAN: 11:49 Yeah. At a high level, the way I would think about what we're building is as an operating system for AI. So, what does the operating system for your computer do? It gives you a bunch of services that every computer user and developer needs to really both develop but also run and facilitate running lots of applications simultaneously. We do similar things for AI. We give people tools throughout the development process to accelerate various aspects of what they're doing. And more importantly, allow them to reason about model development using higher level concepts than they might use in conventional tooling. The analogy to programming is, instead of programming with maybe punch cards or assembly language, we're letting people use higher level descriptions of what their models are and manipulate those. And under the hood, the system takes care of making that go really fast, supporting lots of users and taking care of auditability and so on.

CRAIG: Yeah. In your or [blog posts](https://determined.ai/blog/neural-architecture-search/) and Ameet's talks you talk a lot about hyperparameter optimization or neural architecture search. Is that on the research end or does the platform have that capability built into it?

EVAN: 13:04 So, yeah, very much the platform has support for both of these things built in, and is a central feature in our product offering. I'll let Ameet talk a little bit more about the idea there. But really this comes back for me to the fact that essentially today developing machine learning models is a little bit like throwing spaghetti at the wall and seeing what sticks. It's a highly experimental process. Nobody knows, a priori, how a neural net is going to work for this particular problem on this particular data set and what type of dropout and regularization to use and so on. And so, the solution unfortunately is ‘boil the ocean’ and try every possible combination. And what we give people the ability to do is specify in a high-level language and declarative interface, ‘these are the sorts of models that I'm interested in trying out, this is the resources that I have available to me, and this is the budget that I want to associate with running a particular experiment that might include training thousands of different models and then picking the best ones.’ And underlying that high-level interface and workflow is some of the algorithms that Ameet and his collaborators have been developing for the last several years.

CRAIG: Things like [Hyperband](https://blog.ml.cmu.edu/2018/12/12/massively-parallel-hyperparameter-optimization/), or there was another one, or maybe the same thing, [ASHA](https://arxiv.org/abs/1810.05934).

AMEET: 14:24 Yeah, they're related to each other. Absolutely. I'm happy to explain more about how those methods work and if you think that'd be interesting. But before I do that, one high level point that I wanted to make is that while hyperparameter optimization and neural architecture search are certainly first-class parts of our product, we also think that there's a lot of different types of workloads that ML engineers need to be performing. And whereas hyperparameter search is one of the key ones, there's other things as well. And, right. You mentioned earlier that folks are on their own building their own infrastructure if they don't have access to this unified platform. The way they're doing it is really with duct tape. They're saying, here's this open source thing that does hyperparameter tuning. Here's this other thing that does distributed training. Here is some open source tool that's completely incompatible with each of these other two things, but might help me with reproducibility and literally let me take duct tape and glue them all together. And Oh, by the way, the people who are doing this, their core expertise is in machine learning, not actually in building distributed systems. So, it's one of these things where it’s just bad news all around.

CRAIG: 15:27 Yeah. Yeah. And the point you make as well is that there's a shortage of machine learning engineers out there. So, the skill level is dropping.

AMEET: Absolutely. These aren't people who lack talent or lack very desirable skills. It's just that, the very desirable skills that they have, it's hard for them to unlock them and really leverage them because they're spending a lot of time working on other problems that are also very challenging but for which they are not experts.

CRAIG: Yeah. Yeah. And this problem is growing exponentially, I would guess, as ML moves into industries throughout the economy. I mean industries that are not tech centric.

AMEET: 16:06 Oh, absolutely. What you see is that within the tech world there's understandably this hubris of, ‘I want to build it myself whether or not I have the capability of doing it.’ It's this idea that, ‘we have engineers, we're a first-class engineering company. We're going to build everything ourselves.’ Whether or not it's a good idea. That's the prior for all of these companies. And that it's a natural thing. For non-tech companies, they know that they have a problem and they're really happy to not be building the stuff that's not their core competency. They don't want to be distracted by managing huge amounts of software when their core competency is in some vertical and it's maybe unrelated to building software.

MUSIC: INTERLUDE

CRAIG: 16:55 Since you both have worked on hyperparameter optimization and that that's a core part of your offering. One question I have is, deciding on what optimization algorithm you're going to use is itself a problem, right? In some of the talks I've seen you give, Ameet, you go through all these different strategies and you're gauging the success of Hyperband, for example, against DARTS, but for someone coming from outside and looking at just the academic research, how do you decide which one to use?

AMEET: Yeah. I'm happy to answer that. There is certainly a lot of work in this area. I mean, hyperparameter optimization's not a new problem. It's a problem that when I was in grad school, and I'm sure Evan had to deal with it all the time as well, it's a problem that anytime you're doing applied machine learning you deal with. But since 2012 or so, it started to pick up in terms of research interest and practical need because the amount of data, the amount of compute, the complexity of the models have all grown so much that you can no longer do a hyperparameter sweep over lunch on a single machine on your laptop. Now it can take, I think in some of our talks we talk about, to create a plot in an academic paper for some benchmarking task can take 29 years [in GPU hours] and millions of dollars or almost millions of dollars, right?

AMEET: So, the, the amount of time and money is just is growing astronomically. So, that's the motivation for a lot of research coming out about this. And, Hyperband, it's really taken off. It's been adopted in a lot of places. People are excited about it and I think there's a few key reasons for it. So one, obviously people want something that works. They want to be convinced that using this method is going to get them the most accurate model. Right? But they also want something that is general purpose and robust. So, it's not something where, ‘Oh, it might work for this problem, but it might not work for that problem.’ There's 18 different hyperparameters of the algorithm itself that need to be set in order to get it to work, right?

AMEET: 19:00 So it has to be easy to use and has to be robust and be general purpose. And third, it's really nice if there's some understanding theoretically, some certificate of, ‘not only am I telling you that I think it's general purpose or it's simple or that it's robust, I can provably show to you that it's actually the case and here's why.’ Right? And so, this method, Hyperband, doesn't just work well in practice on the data sets that we've tried, but we've theoretically characterized its behavior, its strengths and its weaknesses. And basically, what we're able to show at a very high level is that, in optimistic cases, it can do significantly better than standard baselines. And in the worst case, it's pretty much as good as those baselines. So, there's a huge amount of upside to it and very, very low risk of it actually not performing reasonably. And so, there's this notion of, it's robust, it's safe, and it's also highly accurate.

CRAIG: 19:50 And so, is Hyperband the optimization algorithm embedded in Determined AI's platform? Or do you have a choice of using Hyperband or some other …

EVAN: So, right now you have a choice and it is the default option that we recommend to everybody. But if you want to run grid search or do you want to run random search or you want to run population-based training, those are all things that you can do within the context of the platform. And I would say that sometimes, when you're doing something like grid search, you're not actually trying to solve a hyperparameter tuning problem. You're actually trying to solve maybe a different problem around ‘how do I ensemble lots of different models together that are representative of different subsets of the data’ or other things. And so, it's not like it's a tool that we think people should completely ignore or stop using. It's just, if you're trying to solve more directly an optimization problem over a surface that is not differentiable and not convex, et cetera, then we think Hyperband is the right tool. And for many hyperparameter optimization problems, that's the nature of it.

AMEET: 20:57 So, I agree with everything Evan said. The one thing I would point out is that it's not just vanilla Hyperband that we have in the product. And I think this is another interesting point where machine learning and systems naturally intersect in a nice way in the sense that Hyperband itself, If we talk about why are people excited about it, it's theoretically principled. You can write it down in pseudo code in eight lines of code. It's very intuitive and simple. That said, when you actually want to build a system that leverages a method like this, there are some system subtleties that are not obvious when you're just thinking about the problem of hyperparameter optimization in the abstract as a machine learning researcher. But, when you think about systems constraints and multitenancy and, various notions of reproducibility and stragglers, things like that, there are variants of the algorithm that make a lot more sense in practice. And so, the high-level method is certainly derived from Hyperband as a starting point, but there's a bunch of bells and whistles that are very much motivated by practical systems constraints, which I think is pretty interesting in and of itself.

CRAIG: 21:56 Yeah. It occurs to me that for the benefit of listeners who don't know what Hyperband is maybe you can give a quick thumbnail.

AMEET: Yeah, yeah, sure. So, this problem of hyperparameter optimization, very broadly speaking, the idea is that when you want to train a machine learning model on data, you typically don't train a single model. You start with a model family. Say you start with a neural network where there's a bunch of different designs, decisions or knobs that you can tune. And so that specifies a whole range of possible models that you could train on your data. And the goal of hyperparameter optimization is defining particular values for these knobs, or settings for these nobs, that result in a model that's accurate enough, that gives you the result that you want. And so, the standard way to do things, the simplest way is something like random or grid search where you start with the search space.

AMEET: So, this is something that you have to define pretty much always for hyperparameter optimization. You start with the search space over different models and you basically randomly sample different points in the search space. For each of these points, you, in a black box fashion, train the model to completion to have some way of measuring its quality or its accuracy and you do that 10 - 100,000 times and then you pick the best out of the ones that you sampled. And that's the result of your hyperparameter search. The idea of something like Hyperband is, or the issue with this approach. and it's a very similar issue with somewhat more recent methods that are based on Bayesian optimization where you don't passively sample different points from the space, but in a more iterative active fashion, you want to sample different points in the space.

AMEET: 23:29 The problem with both of these approaches is that as you have more and more hyperparameters, the space that you're searching over gets bigger and bigger. But it turns out the space gets bigger exponentially. So, there's this thing called the curse of dimensionality, which means that the volume of a space grows exponentially with the dimension of the space. And so, working with two hyperparameters, it's pretty easy to cover the space, try a bunch of options and hopefully get a good one. But if you have a 10 dimensional space, it's exponentially harder than a two dimensional space. And so, the point is that to do things in a reasonable amount of time or a reasonable amount of budget, you need to change the rules of the game. And so, with something like Hyperband, it says, okay, well I can't do what I was doing before.

AMEET: 24:09 Instead I want to consider more points in the search space. But to do that, I'm not going to just, in a black box fashion, evaluate the quality of each of these points. Instead I'm going to change the rules and make an assumption that each of the points corresponds to a model that's trained, say, in an iterative fashion using gradient descent or stochastic gradient descent, which is how the vast majority of models are being trained these days anyway. And the point is, once you make that assumption, you can then, instead of deciding to train every model to completion, you can do a competition or a tournament, right? So you start with a bunch of different models. Let's say you start with 64 models and then instead of training them all to completion, you train all of them a little bit and then you keep the top half and so now you are down to 32 and you train them all a little bit more and you keep the top half and then you're down to 16 and so on and so forth until you get to a single winner.

AMEET: And so, the point is that you for the same total amount of work, if you consider work the number of training iterations across all models, you can allocate your resources adaptively to configurations that seem more promising. So, the very high-level idea is to peek inside the black box of training a particular model and leverage early stopping. And then the hard thing is, how do you do this in a robust, safe way. Because if you stop too early, you're not going to have enough signal to know what's good and what's not. If you stop too late, you're being too conservative. And you don’t want to make too many assumptions about when you should stop because if your assumptions are faulty, which they often are going to be, then you're kind of again, you're in a bad situation. So, the devil is a little bit of a detail about how to do this to get the empirical gains, but to provably and generally also make this work.

CRAIG: 25:46 Is it fair to say that it's a strategy, a systematized strategy for early stopping?

AMEET: Yeah, absolutely.

MUSIC: INTERLUDE

CRAIG: And what, what are some of the other components then that that the platform can perform? I mean, within the hyperparameter optimization, there's a neural architecture search and I heard you, Ameet, talk a lot about the failings of neural architecture search at this point. But does the platform, have that functionality built in?

AMEET: 26:22 Yeah. So, there's been a lot of buzz and marketing around neural architecture search as this fundamentally new and different problem that no one's ever really considered, when in reality what it is, is a very interesting problem. But it's really just a specialized part of the broader hyperparameter search problem. Right? So, it's still this idea of, I have a bunch of different neural architectures that I want to be considering and I want to define a space of architectures and I want to find a good architecture within that space of architectures. And I want to search through that space as quickly as possible because I have computational or monetary constraints. So, it's still a hyperparameter search problem. The way, in particular, that you defined the search space is specialized because you're dealing with neural architectures.

AMEET: 27:08 So it's a particular type of hyperparameter that you're dealing with. But the point is, and we wrote a recent paper about this, is that while it's an interesting problem and while there have been some interesting advances that have come out of it, the field is just the specialized methods for neural architecture search. Again, going back to your point of what methods should you use. Nobody I would argue should be using specialized neural architecture search methods yet. They're not robust, they're not reproducible, they're very specialized to work on one, maybe two benchmarking data sets and when you accurately compare them against baselines, something like a productionized version of Hyperband, even when the specialized methods might be overfitting to one or two problems, this more general-purpose solution that is just run out of the box is nearly just comparable and probably is comparable if you ran it on a few more data sets. So, the gains so far are not, it's interesting from a research point of view, it's definitely something that is worth understanding better. But, I wouldn't recommend it. Anybody who is using it in practice, I would really question why right now.

EVAN: And I think the, the interesting piece of this today is that we have customers that are running neural architecture search as a hyperparameter tuning problem and seeing great success with it. And so, in practice, we're also seeing real gains there as well.

CRAIG: 28:24 So without neural architecture search, it's still based largely on intuition or on past practice?

AMEET: Oh, no, no. So, let me just clarify what I'm saying here. So neural architecture search is a particular problem to be solved. There are then specialized, there's different ways to solve a neural architecture search problem. Now, this neural architecture search problem can be solved by say, Hyperband, or it can be solved by something like, as you mentioned, DARTS, which is a specialized algorithm only for neural architecture search. And what I'm saying is that you can solve that same NAS problem about as well these days with something like Hyperband in a much more general and clean way than you can with something like DARTS. You can still solve the problem. The question is what tools do you use to solve this problem? And the specialized methods are immature, not well understood and not robust, not reproducible, and as a result, not ready to be used. If you have a NAS problem, you should be using something like Hyperband or some other mature hyperparameter optimization method to solve it. Not a specialized NAS method.

CRAIG: 29:27 Yeah. Yeah. One of the things that interested me in the platform - and we can talk about the research behind the platform as well if you'd like - is that it records the meta-hyperparameters so that you have an audit trail. Is that right? And so later when something is deployed and there are products in the real world and if something goes wrong, you can go back and see what changes were made or why certain decisions were made or am I over optimistic about the functionality there?

EVAN: No, no. So, from a product perspective, the ethos is let's capture everything associated with your model development life cycle. So that's every line of code that goes into your model definition. What version of the data were you running this particular model on? What were the hyperparameters, what were the random seeds? What were the versions of the various libraries and the operating systems and so on all the way down the stack. And we leverage tools like Docker and so on to help make this a reality. But one of the fundamental tenets there is that first we want to enable reproducibility across the various training cycles of these models so that if I run the same model on the same data today as I did yesterday, I should get the same model parameters out.

EVAN: And you'd be surprised how hard that is to do with modern machine learning toolkits. We have a series of blog posts explaining in excruciating detail what we had to do to make this work. But, the focus there is that if we can enable this level of reproducibility, we can provide these sorts of audit features that you're talking about. But we also foster a new style of collaboration among machine learning engineers and data scientists so that as your team starts to scale, now you've got this system of record that keeps track of every model you've ever trained, the accuracy that it got on this particular task at this particular point in the training life cycle, how it relates from a code and data perspective to some of the other things that you've trained.

EVAN: And you can begin to wire that up to your deployment and serving system in such a way that you keep track of when this model starts failing. What were the sets of data associated with that particular thing? What didn't I see? And while right now the focus of the product is really on that training side of model development, ultimately, we fully expect to be able to complete that data loop in terms of collecting data from production usage and feeding it back into the training system in a really robust and systematic way.

AMEET: 32:06 I think again, here, this is another example of machine learning and systems, the intersection of them being important, right? From an ML point of view, for a large part we've been the wild, wild west. We've been optimizing for accurate models, right? Everything we do in terms of reproducibility - there's a bunch of models that large companies have in production that they can't reproduce and they know that they can't reproduce it because they can't reproduce them when they're getting stale or not working that well. They're very hesitant to pull them because they don't know how they're going to replace them because they don't know where they came from in the first place. And a second point is that my own experiences, and I think it's pretty common, is that you do your best with reproducibility, but you're responsible for saving all your log files and naming them in ways that are unique and hopefully meaningful to you six months from now when you need to look at them.

 And you're hoping you're capturing all the information, but again, your real goal is to get a good number. It's one thing when this is an immature field and we're getting started. It's another thing when it's powering some of the biggest organizations in the world. This needs to be more of an engineering discipline. And you think about typical systems, reproducibility is a core tenet of most systems, right? So, this isn't a new idea in the systems world, but it's something that we need to learn, us ML folks, from the systems community in terms of how to make this a first-class property.

CRAIG: 33:26 Yeah. And that raises the question, because of course reproducibility is a huge topic in the research, in the academic research world because so much of what's presented at conferences is with inadequate code and it's never reproduced. Is it possible that this platform could be adopted at universities so that there is a standard and a clear audit trail and reproducibility will be much easier because you've got everything in one system?

AMEET: Yeah. Yeah, I think absolutely. And I think a related point, it's one thing to say that you have a system that allows you to capture everything, but, a very equally important part of that system is that it doesn't get in the way of the user. It's seamless. It happens behind the scenes. So you as a user can still focus on optimizing for accuracy. But behind the scenes of that, without really doing much of anything, all that information is being stored for you. So, and, from a research point of view, people aren't actively trying to cheat in any way. It's more that they're focused on numbers and sometimes they might make mistakes and not capture those mistakes and therefore can't reproduce them. So, if there is a system that allows them to reproduce things that doesn't have much overhead associated with it, then absolutely I think these sorts of ideas are absolutely things that should make their way into the research community.

CRAIG: 34:44 And on your platform, you simply duplicate a model that you find in the archives and that drags with it all of the stuff that you might not think about it.

EVAN: Yeah, that's right. And of course, you can't have perfect control, right. If somebody decides adversarially that they really want to introduce a new source of randomness that we don't capture, that's not something we can control. However, if people use the system in the way it's intended and don't go out of their way to try and fool it, this audit archive gets built, as to Ameet's point, really naturally, as exhaust of the research process.

EVAN: 35:26 I think some people look at this problem of reproducibility in ML and they say things like, ‘Oh, if you just taught the machine learning practitioners how to use Git or some source control tool, all of this would be solved.’ And I don't think it's quite that simple because of course there's data and there's randomness and a bunch of this complexity. But it's also the case that when I'm writing a traditional piece of software, usually what I'm thinking about is, okay, I have an idea of how to implement this feature. I write these 30 lines of code, I add them to my source control, the code is done. When I'm building a machine learning system, I have no idea which 30 lines of code I need to write. And so I'm constantly experimenting and trying new things and it's very hard to convince myself that I should be recording every possible thing that I had tried in the past simply in order to pick the one that works. Whereas, if you turn it into a way where all the things I've tried have been captured as exhaustive of the work that I'm doing, it becomes much more natural and seamless for ML developers to use it and then go back and retroactively say, what was the important thread that I worked on, all these other branches, they didn't work out.

MUSIC: INTERLUDE

CRAIG: 36:40 There's some other things I wanted to touch on before we move away from the platform, I mean, it is specifically an AutoML platform, is that right? Or, or can you use it for a hand-crafted model.

EVAN: Yeah. So, we absolutely have customers that are using it for handcrafted models. Machine learning model development really is a process and there's a life cycle of a model, right? In the early days, you're just trying to experiment and see if you've got any signal whatsoever. And before, you find something that works. So, you're trying maybe baseline methods and linear regression and so on, on your models or, baseline CNNs if you're doing image classification for example. But then once you've convinced yourself not only that there's some signal in your data and that this is something where modeling can help, then you get into this phase of what you'd like to be rapid experimentation where you're trying lots of different ideas out about what works and what doesn't. And this is where tools like Hyperband come in and can accelerate things.

EVAN: 37:43 But throughout that life cycle and then once you get past that, there's this question of how do I push this model into production? How do I make it compressed enough to run on my mobile device? And so on. And the way I think about our platform is we are providing tools that can be used at various stages of that model development life cycle, automating as much of it as we can, but also giving our users the ability to do the handcrafted things when they want to. So early on in the process, let's say you're training a CNN on a big dataset, what's a tool that you might want? Well, you might want to be able to leverage lots of GPUs in parallel to train your model faster, right? So, you want to get to an answer much more quickly than waiting a day or a week or longer for your models to converge.

EVAN: 38:27 You'd like something that gets you an answer in an hour or two. One of the things that we do on a platform perspective is automatic parallel and distributed training. So, letting users specify declaratively what model architecture that they want to go train by hand maybe, but then automatically taking care of the parallelization for them in a way where it's not like I have to write 10,000 lines of parameter server code to get this to work. It's more, I change a line in a config file and the system figures out the way to do it.

AMEET: Another answer to that question, it's a very different answer, is that I think it depends on how you define AutoML. And I think AutoML right now has been conflated with hyperparameter optimization or neural architecture search. And I think that's looking under the light posts. Those are two problems that are clear, concrete, tangible problems that you can specify quantitatively and therefore it's very mapped and they're the problems that people understand and feel the pain of today and therefore it resonates with people and it makes sense to study them in the short term.

AMEET: But the AutoML is much, much broader than hyperparameter optimization or neural architecture search. And, without repeating everything Evan just said, all those things that Evan just said, some of these problems are more open-ended. I can't specify them as a math problem just yet. But everything from collecting my raw data to deploying a model and monitoring it and everything in between, that all should fall under the umbrella of this broader AutoML problem. And so, in some sense, yes, you can certainly use the product without doing hyperparameter optimization or neural architecture search. But we think that everything from beginning to end should be more automated and therefore everything in some sense is an AutoML problem. Does that make sense?

MUSIC: INTERLUDE

CRAIG: 40:20 You used, in many of your talks, this analogy of the pioneer age of aviation compared to the jet age. I would say that we're in the propeller age. I would suggest we're beyond the pioneer age, but maybe not…

AMEET: Yeah. The way I tell that story is that the pioneer age is the Wright brothers at Kitty Hawk in 1903 and the few years after that. The next point in that timeline is the Wright brothers and other folks demoing all the advances over those last seven years in 1910 at the World's Fair and then the 1950s as the jet age when there's commercial and broad, broad adoption of this stuff. And I would argue that we're at the World's Fair moment right now. In 1903, it was science fiction, people did think that powered flight was necessarily a real thing. The Wright brothers, even though their actual experiment was very modest in absolute terms, demonstrated that this could potentially work in real life.

AMEET: Right? It was no longer science fiction or fantasy. Fast forward to the World's Fair - the World's Fair, I wasn’t alive at the time and I don't think is today what it was back then, but I think it was a big deal back then - and demoing at the World's Fair really signaled that this was a cultural phenomenon that people realized was going to change the world. Right. And I think that's where we are with machine learning or AI at this point, right. People don't think it's just hype. There's clear, tangible successes we've already seen with all this stuff. But at the time of 1910, you have to be really rich or be fighting a war to actually be flying planes. And, I don't know about the war aspect of machine learning right now, but, there's the Googles of the world who are able to use this and then there's everyone else.

CRAIG: 41:58 And AutoML and Determined AI's platform is part of the process of streamlining and productizing the process is that right? So that research people focus more on discrete problems rather than spending their time building these complex models. And then on the industrial side people can adopt proven solutions much more quickly and productize them.

AMEET: Our square focus is on industrial applications of AI. I think that the fact that we are closely tied with academia and academic research allows us to sift through the new innovations that are actually ready for prime time versus those that are, while interesting, maybe not quite mature enough to be adopted more broadly.

EVAN: 42:55 To put it another way, in with the propeller analogy, probably you had about a hundred papers a year coming out about new propeller designs, right. And which was going to make optimal thrust and make the planes go faster or whatever. We're really good at knowing which of those propellers should go into the next generation of planes that are being built. And that gives us a way to accelerate the right technologies into the hands of our customers.

CRAIG: To get to the jet age then, to continue with the analogy, is it a question of better algorithms. Is it a question of better hardware? Is it a question of larger datasets?

EVAN: I would say all of them for sure. Right? You need algorithms that are not just better and faster, but also safer.

EVAN: Right. And reliable and Ameet touched on this earlier, but I think that that's a central tenet around what we're building. And we don't necessarily think that people should just be optimizing for accuracy. Maybe some notion of fairness or transparency should be embedded in the objective functions that they're going after. We want to let people do that in a really flexible way. Data is like the fuel, right, for these models to really go. And every time data volumes go up by an order of magnitude, something special happens. It's not totally predictable. You can't just throw more data at the same exact model and expect it to get better. It will marginally, but what we see is once you have more data, you can start to do more complex and more interesting things with the models than ever before.

EVAN: And hardware is super important. We're already pushing special accelerators to their absolute limits in order to run convolutional neural networks as fast as possible. And it still takes hours or days to get these things done. And with the death of Moore's law, there's going to be an increasing number of specialized chips that hit the market that are going to do things like help us train deep learning models faster or do inference faster in power constrained settings. And so, I think from a software perspective, all of that is going to lead to increased complexity for the end user who's developing these machine learning models. And so, where we come in is, we think that our system can help intermediate much of that complexity and hide it all under the covers from folks.

CRAIG: On safety, and particularly on bias but as well on safety, on robustness of systems, that's an issue with the algorithm. So, it's not something that AutoML by itself is going to solve.

EVAN: It's beholden to the user to express that constraint in the context of their modeling decisions, right? So, I mean, what am I actually optimizing for if I were trying to reduce fraud in bank transactions to zero? What's the simplest way you do that? You don't let there be any bank transactions, right? But that's not right. That's not what we want there to be. We still want there to be transactions, just below a certain rate. And so, somebody who's designing that model has to express and there has to be at least so many dollars that go through the bank every year or something like that. And I think the same analogy can be made for bias. It's not just, I need to validate loan applications in a way that's going to maximize profit for the bank. It's very different. Instead, we have to take into account these notions of bias and fairness. And it goes beyond the important issue of our social and cultural definitions of bias and gets into the weeds of any particular domain. And so again, I think it's the responsibility of the modeler to figure out what they really are meaning to express when they're using machine learning to solve their problems.

AMEET: 46:42 And, I would add to just say that, right now, or up until recently, people have been optimizing for exactly one thing, which is accuracy, right? Forget about everything. Forget about cost, forget about privacy, forget about fairness, forget about bias, forget about interpretability or transparency. All I want is accuracy. There was this recent paper that came out on green AI, which, and this is more focusing on the research community, how there is this barrier between the haves and the have nots. Those that have the ability to throw more computation and more data at a problem. And that's obviously just simplifying it because the people who have access to more data and more computational resources are also quite innovative and making use of them and actually getting new, better results. You know, doing well on accuracy benchmarks.

AMEET: But, that's where we are today. This green AI paper talks about how we need to focus on efficiency more. That's something that we at Determined AI focus a lot on as well, which is the case that maybe if you're at Google you have the ability to run an experiment on a seemingly boundless number of GPUs. Any other organization they have hardware constraints, right? We all have constraints and it's not just how quickly you get an answer. It's what is the tradeoff between taking using half as many resources. Does it take half the time or does it take only 90% of the time or, what is the scalability, what are the speed ups that you're getting there?

AMEET: But then when you think about something like fairness or privacy or interpretability, that's, as Evan said, yet another metric you need to be optimizing for. But I would argue that these are things that we don't really know how to do yet. Right. It's a great thing to see that this is starting to be a very active area of research, but it's not necessarily a mature set of research. We're still coming up with the different definitions of fairness. The different definitions of privacy, different definitions of interpretability. And, a lot of times these definitions are in conflict with each other. It's not clear which ones apply for different applications. Um, and so we as a research community are still working through those issues. Longer term. Absolutely. These things need to be taken into account. We have a benefit right now that current ML engineers are pretty advanced users. So, if we endow them, we give them all of the metadata and all of the all of the raw metrics associated with all the experiments that they're running, they have the ability to sift through that data and as experts figure out some of these questions on their own in a more manual way. But, over time that should be automated. And so, you could argue that maybe that does fall into, again, a much broader notion of auto ML.

CRAIG: What you're saying is a platform like yours that captures all of this metadata that might not be captured otherwise creates a pool of data that then can be used to optimize for fairness at some point, if there are strategies that eventually emerge.

AMEET: Today it's the best we can do, is maybe what I'm saying. I guess what I don't want to say that we and nobody else has solved the problem of how to make machine learning fair, unbiased, or perfectly private. And that's not a shortcoming of us as researchers or us as a company. It's just the current state of the entire field of machine learning. It is something that is increasingly important that we're aware of. We are comforted by the fact that the users that we work with today are pretty sophisticated and they have some understanding of these ideas. But over time this needs to be more automated and more robust.

CRAIG: How then are you pushing this out to industry at this point? And is the uptake strong?

EVAN: 50:09 So we work really closely with what I'd say are a sophisticated set of customers, where we're talking about research labs at, say, a major ad tech company where there are a hundred AI engineers who are all working on deep learning models day in, day out, and they have loads of GPUs in their big cluster; at autonomous vehicle companies, where they're building state of the art computer vision models, again, to increase safety in autonomous vehicle applications; at places like pharmaceutical drug discovery companies and gene sequencing firms and so on. And so, these are a broad array of use cases for deep learning and AI across a bunch of different industries. But the common thread is, one, they tend to be sophisticated users, people who live in tools like TensorFlow and Pytorch.

EVAN: And who are experts in their domains, whether it's computer vision or, or genomics or, or financial data for example. But who maybe need the power tools that are going to enable them to be much more productive than they are today at developing these solutions and bringing them out to market. And so, one strategy would be the default thing that everybody who reads a Keras tutorial on Hacker News goes in and reaches for it. That's an interesting market. But, instead we look at where the value is being created with AI in an industrial setting today and it tends to be out of these more sophisticated groups. So that's who we go to help first with the idea, of course, that we're going to be able to assist them and learn how they like to work and take those workflow patterns and styles to the broader market over time.

CRAIG: Yeah. Because that broader market is, although these very sophisticated companies are doing interesting things for the economy, the excitement is optimizing, cardboard box manufacturers or shoe makers or anything. Every company can apply ML at some level.

EVAN: 52:15 Yeah. And we're seeing uptake in our product from essentially the places you'd expect: Industries where they are already very data-driven and are used to using statistical models in some way or another to make decisions. And this is the next step in their evolution to being even more data-driven and more automated. Now, I think, because these companies are having such success and in many cases knocking over incumbents in their industries and becoming the new de facto standard, you're going to see a lot of more traditional companies understand that they either have to develop this expertise and get up to speed with it or be bowled over by the next startup. And tooling to enable their relatively hard to find small teams of data scientists that they can manage to bring on today, to make a team of five behave like a team of 50, that's really high leverage on those people's time. And that is what I think we'd like to provide.

CRAIG: 53:12 Yeah. And looking forward, is the long-term vision that your platform becomes a standard in the way that Salesforce is the standard now for CRMs? That everybody who's building a machine learning model without a lot of expertise looks around for a productized workflow that they can adopt. And the common wisdom will be Determined AI?

EVAN: In the same way that everybody up until maybe 15 years ago thought Windows was the going to be their operating system of choice, I'd love for Determined AI to be their AI operating system of choice. Yeah.

AMEET: We’re working with advanced users today. We think that for every user we have today, I don't know if there's going to be 10 or a hundred or a thousand of them in the next few years, but this idea that we are at this point where, there's cultural acceptance, but there is about to be this wave and massive rush of new demand and companies are screaming that they need more employees who are educated in these different areas. And, very soon we're going to see orders of magnitude more ML engineers who are orders of magnitude less sophisticated about what they know about and we need to make them productive.

CRAIG: 54:25 That’s it for this week’s podcast. I want to thank Evan and Ameet for their support. For those of you who want to learn more about their platform, check out their website at Determined.ai. You can also find a transcript of this episode on our website: Eye-on.ai. Thanks again to our listeners.

 The singularity may not be near, but AI is about to change your world, so pay attention.