This Week in Machine Learning & AI: Scott Clark Interview

Introduction

Sam:
Hello and welcome to another episode of twiml talk the podcast by interview interesting people doing interesting things and machine learning and artificial intelligence. I'm your host Sam charrington this past week the conference finally came to me over the weekend the great and not so little anymore strangeloop conference Grace downtown st. Louis. I got a chance to meet with a bunch of the speakers including Suman chintala of Facebook Allison Parris of NYU and Sam Richie of stripe. I had a ton of fun and I can't wait to share some of these great interviews from the conference before we move on to the show speaking of conferences. We're going into conference give away mode for a few days next week, October 10th through 11th. I'll be in Montreal for the rework deep learning Summit. And one lucky listener will get a chance to join me. They're entering the contest is simple just head on over to twimlai.com DL Summit and choose any of up to four methods of entry and Wawa while they're only four ways to enter this time around by sharing the contest with friends each participant can get up to 14 and trees. This giveaway will only be open until noon Central Time on Wednesday the 4th. So make sure you get your entries in ASAP. Good luck this week. I'd like to introduce a new sponsor necklaces and thank them for sponsoring this week show nexosis is a company of developers focus on providing easy access to machine learning. The next Solstice machine learning API me to developers where they're at regardless of their Mastery of data science. So they can start coding up predictive applications today in their preferred programming language. It's a simple as loading your data and selecting the type of problem. You want to solve their automated platform trains and select the best model fit for your data and then outputs predictions get your free API key and discover how to start leveraging machine learning in your next project at nexosis twiml. That's n e x o s i s t w i m l head on over check them out and be sure to let them know who sent you finally before we dive into the show or reminder about the upcoming twiml online meet up. On Wednesday, October 18th at 3 p.m. Pacific time. We'll discuss the paper visual attribute transfer through deep image analogy by jingle and others from Microsoft research. The discussion will be led by Duncan suthers. Thanks Duncan to join the meet-up or to catch up on what you miss from the 1st to meetups visit twimlai.com Meetup as you all know a few weeks ago. I spent some time in San Francisco at the artificial intelligence conference by O'Reilly and Intel nervana while I was there. I had just enough time to sneak away and catch up with Scott Clark co-founder and CEO of sick opt a company who software is focused on automatically tuning your models parameters through Bayesian optimization. We live pretty deeply and to how they do that through the course of this discussion. I had a great time and learned a ton but be forewarned this is definitely a nerd alert show. And so without further Ado onto the show everyone I am here with Scott Clark Scott is the founder and CEO of a company called stopped and he was gracious enough to spend some time with me this morning to talk about his background the company and the topic that I am very interested in learning more about Bayesian optimization. We're sitting in his in his office in San Francisco. I happen to be in town for the AI conference and I'm really looking forward to this interview so welcome Scott. Thank you so much really looking forward to it as well. Awesome. Awesome. So let's just jump right in and have you tell us a little bit about your background and how you got involved in machine learning different one. So, First got really excited about this while I was in grad school. So is it pursuing a PhD in applied

(end of excerpt)

Interview

Guest:
math at Cornell University?

Sam:
Go upstate New York. Yeah exactly

Guest:
when I was great because there's not a lot to do and it's super bad weather all the time. So you just focus on studying and you graduate as soon as possible.

Sam:
I went to RPI undergrad also Upstate New York and had the same experience

Guest:
highly recommended efficient degrees

Sam:
RPI have the added advantage of it was hugely skewed towards now students. And so there were even less distraction

Guest:
and that's excellent. So basically, I was applying math to a variety of different things. One of the focuses of my degree was bioinformatics. So I was my I had a fellowship from the Department of energy. So the problem that I was trying to attack was genome assembly and you can think of this as trying to solve a jigsaw puzzle on a supercomputer, right? Basically, we have a bunch of the DNA. We have to reassemble it into some jeans in the department of energy cares about this because if you know the do you know it might be a path towards more efficient biofuels or something like that. The problem was lots of tunable knobs and levers with these various systems and we had to configure those to get the best possible performance out of that

Sam:
and you are the grunt in grad school that like to know all these lower. We

Guest:
jokingly call this graduate student descent the idea of being we just need to get to the best configuration and it doesn't matter how you get there. Yeah, and so the standard way to attack this problem and a lot of interviews value for your paper something like that. I mean I can make incentives are a completely different. Topic. Yeah, I

Sam:
just thought I thought that

Guest:
but the idea is there's a couple standard ways people go about attacking a problem like this either try to brute-force the problem. So just lay down a grid of all possible options for every configuration and try to write this was intractable for us because it took 24 hours on the government supercomputer every single time. We

wanted to try single

Sam:
configuration

Guest:
random. I search just become very popular, especially in the Deep learning literature. We're trying to come up with different configurations of hyperparameters and architectures and things like that turns out much more efficient than go in search but this is still like trying to climb a mountain by jumping out of an airplane and hoping the way into the paint not necessarily the most intuitive way to go about

Sam:
optimizing something a lot of the different our rhythms. We use randomized initialization that's different from random I

Guest:
search correct. So when you're building a neural network, you might use randomized initialization on the individual weights and then use some sort of stochastic gradient. Send Optimizer within that underlying system. This is more of a black box Time an optimization problem on talking about what we're not interest specting the underlying model, but just tuning the higher-level configuration parameters. So some of those configuration parameters might have to do with that random initialization or the stochastic gradient descent parameters or something like that. You definitely need to be able to bootstrap efficiently from no data, but doing purely random I search is not Australian most efficient thing to do.

Sam:
So maybe before we move on I think we're going to be spending a lot of time talking about I prefer murder optimization theater, you know, maybe dig into research a little bit more so that we're all starting from the same place. Basically as I understand it the idea is you've got some set of hyperparameters, you know, those form and an n-dimensional, you know, n-dimensional not a cute plus a lotta. Thank you. And you know grid search is basically systemically, you know going from point-to-point like if you were searching for someone in a forest you kind of form a grid and kind of attack all those ones. Yeah. Well and random is you're basically picking points and the ideas statistically if you pick it up when she gets some level of coverage of you know, all of the combinations of these hyperparameter exactly.

Guest:
So back to your search is going to force analogy. This is yeah jumping out of a helicopter and seeing if the person's they're continuing to do that over and over again. That's right. That's right of my search. Another popular method is just manual to it. So trying to do this in your head and enforce example, when there's only two Dimensions you might have a lot of intuition about maybe the person's going to be up on a hill or something like that. It can actually be somewhat effective. But once you start to look at 20 dimensional problems a lot of human intuition starts to break down and you might not be able to have some of that expert knowledge in the searching for a human and a forest setting how to set Casa gradient descent parameters and number of hidden layers and learning rates and all these sorts of things that starts to get very convoluted very quickly and so many will search. Well, it can be effective to kind of resolve very localized Solutions is not a great Global optimization strategy.

Sam:
And for the typical model that you are seeing like how many high performers are there? Yeah. So it really

Guest:
depends on the underlying systems something simple like a random forests not only have a couple of each other about number of trees number of samples needed to split a note something like that as you start to Advanced maybe the gradient boosting methods and I'll assume you have learning rates and other sorts of parameters you can tune. But once you get into the deep learning and reinforcement learning machines that can be dozens of individual parameters, especially if you start to think of the system as a whole so when are you doing an NLP or computer vision type problems all sudden you have different ways. You can try Matt rise the data as well and so by looking at that system and its Entirely all of a sudden there can be dozens of parameters and something that grows exponentially like a grin sir. Just completely intractable the human manual intuition starts to break down and ran to my search is just too slow to luck into a reasonable solution. Okay.

Sam:
Can you give an example of in the case of NLP how the the way you look at the data set changes and increases your apartment or space? Yeah. So how you tokenize the

Guest:
text itself? So if you look at different engrams sizes, the idea being do like a one word at a time pairs of words triple got word. You maybe do different threshold for the frequency within the Corpus itself. So maybe cut out words like because they're too common and then also cut out words like Bonanza because they're too rare right? And so you can kind of change the actual future representation itself before you feed it into the machine running over them, but these are all tunable knobs and levers

Sam:
got it. Okay, and so you were stuffed in bread. That's cool white again. Totally need these lovers and you know as all Innovation happens you thought there's got to be a better way because they're

Guest:
so went around the department and found that this was a very common problem people in machine learning people and financial engineering like everybody were building these these expert systems, but they needed to be fine-tuned. But everybody was using these kind of standard techniques so expanded my search outside the department and eventually found who would become my PhD advisor in the operations research field. So they've been attacking this problem for decades. If you have a time-consuming and expensive to sample system had you most efficiently get to the best configuration. So this crops up if you're tuning a particle accelerator it crops up if you're trying to decide where to place a gold mine which is where some of the original research came from in the fifties, but it Maps extremely well want to live right and computational problems. Okay. Yep some input that comes in some output that you care about. How do you get to the best output and if you input attempts as possible, so I started working Still the optimal learning it has called in operations research or sequential model-based optimization or Bayesian optimization. A lot of fields have different names for it. But the idea is how do you do this as efficiently as you can ended up hitting my PhD towards working on this problem and ended up being one of the chapters in my thesis and after graduating. I realize that a lot of different people in a lot of different Industries had this issue. So I spent two-and-a-half years at Yelp working on their advertising team applying the same techniques to help do more performance advertising. Okay the idea of being if you think about it mathematically in advertising system is very similar to a genome assembly system insofar as a lot of experts been a lot of time building something there's a bunch of inputs and there's an output you care about and genome assembly. It's better papers. Can you get a better genome in advertising system a bunch of money comes out the other end. I

Sam:

mean clearly there are tons of problem the fitness General exactly. Exactly. And so you started pick up. How long have you And that it here. Yeah, so immediately after

Guest:
Yelp started stick up about three years ago, like a y combinator and winter 15, raise a few rounds of funding and most recently A series. I led by Andreessen Horowitz and now we're 16 people in San Francisco.

Sam:
That sounded like and so I guess you know, I wanted to kind of jump into you know, the the main Crux of this interview which is around the space in optimization, like walk me through, you know the way and you know, folks like Pedro Domingos. So talk about like the Bayesian to like this one tribe within machine learning and you know as opposed to others, you know, kind of walk me through like I guess what I'm trying to get at is like I've had a couple of conversations with folks about you know, different aspects of you know, like Bayesian program learning and other things but yeah, I feel like you know, there's still some you know, they're still like that some details of like what it means to be kind of Bayesian and think about things from that perspective that we haven't fully captured on the broadcast. So if we can like start there and get to the optimization that would be yeah definitely. So

Guest:
the way of one of those other techniques work like Winters are random search is there's no learning happening and I think that's one of the major differences between the Bayesian optimization approach of the Bayesian approach this problem and some of those more traditional techniques the idea of being every single time. I evaluate this underlying machine learning pipeline or whatever it is. It's extremely time-consuming and expensive and I want to be able to leverage that data to decide what to do next. And so a lot of the Bayesian methods rely on this concept of trading Exploration vs. Exploitation. So we want to be able to learn as much as we can about that underline response surface how it varies how old the parameters interact over what length scales how certain we are about specific configurations and how old Uncle Frodo perform and learn about that while also exploiting localized information to drive me to better results and by constantly trading off these two facets were able to exponentially faster than something like an exhaustive grid search arrive at better Solutions and the main difference here is the fact that we're learning from the past and using that influence what we do in the future.

Sam:
And that one I think about this kind of explored explored Boyd trade-off. One of the things that jumps to mind for me is reinforcement learning. Does that come into play here or maybe less? So because the environment itself a problem itself doesn't you know necessarily change in response to the input.

Guest:
So the the underlying system can change pretty dramatically so you can think of this as this larger system that fits around any underlying pipeline that could be a reinforcement learning pipeline. It could be just a standard deep learning or it could be something as simple as a logistic regression or a random forest and you can think about the fact that every single time we try a new configuration. We want to observe some some sort of output at the end that the user to find could be something simple like accuracy could be the the Sharpe ratio of a backtest of an album like trading strategy or whatever it may be and so we use that to kind of influence what we do next you can think of this as kind of reinforcement loop as a whole over that entire system. But we're agnostic to what the underlying messages. I

Sam:
I guess that and so the underlying method is you know could be reinforcement learning or any number of other

things but it also sounds I was I guess what I was asking was are you or could you do you reinforcements learning at the top level to optimize the thing that you're optimizing which could be reinforcement learning as well the reinforcement learning on the high programmer space as opposed to the actual model itself?

Guest:
Yeah, definitely and there's a lot of different approaches to this underlying problem is a lot of very cool papers that are all the top machine learning conferences for attacking this the way that we attack it is the this concept of sequential model-based optimization. And this is a very Bayesian approach and the idea is were sequentially learning as much as we can about this underline system. So once again using the history to decide what to do in the future It's model based in the sense that we're building up different surrogate models for how we think individual configurations are going to respond when we actually sample the underlined system. We can use various different things here like process fees or other account evasion regression type systems and we want to be able to say given what we think is going to happen. How do we sample as efficiently as possible? So then we want to say what do we think is going to improve in expectation the most with the highest probability of improvement in terms of that new configuration to suggest and then that Loops back into the underlying system after you sample it and we learn update the posterior of these individuals surrogate methods optimize on them and repeat that entire process.

Sam:
So how do you get to the kind of this proposed model for the model based P service

Guest:
in general in Beijing optimization usually Take a specific type of model and and go from there so than some of the open-source work. I did at Yelp. It was kind of very cut-and-dry education process is expected Improvement to optimize and go through kind of extremely sequentially. This is very similar to experiment and other popular Library say that one again spearmint. It was an open source Library out of Harvard very similar to the metric optimization in general though, which I wrote it Yelp also similar to like Jeep I opted which is a kind of more recent one is a kind of the the bread-and-butter Bayesian optimization approach auction process is expected Improvement would stick up her presents though. Isn't this Ensemble based approach so different sort of models different acquisition functions different covariance kernels for learning how the primaries in Iraq as well as not just kind of that standard Bill to us a single sequential a surrogate model-based approach but really taking all of these different optimizers and optimizing and and making it automatic So you can select something ahead of time because you know, you want to take a very specific approach or you can take the more generalized approach and say we're not necessarily going to say we're going to use this specific story of model. We want to learn along the way it was the best possible thing for that underlying system that we're optimizing,

Sam:
right? So to take us the back you are in the former case where you're picking a model specific model, you know, let's say we're assuming gaussian distribution, then basically you've got this type of parameter space. We are I'm trying to get at like how you know, so the parameters of your Gap and distribution of the you're meeting your standard deviation and how are we like what's the process for for identifying those that has been you know that we're doing sequentially

Guest:
gotcha. So the way that aggression process works is that it's assuming that the Response of that underlying system that we're sampling is going to be gouging distributed at any given point. So it's not a single couch and distribution or something similar to like a gathering mixture model. What it actually is is an infinite number of potential couch and responses for every potential input. And then the way the government processes are

analytically to find once you start to sample underlying points, you can explicitly build up with that distribution is at sample points or on Sample points the main thing that controls this is what's called a covariance, and what that is is how much information do I get from sampling point a about some other point be so does it decay exponentially? Is there some sort of Hibernians or noise associated with it? Whether the length scales over which all the different parameters interact just becomes doubly complicated once you start to look at heterogeneous configuration spaces in New Jersey. Continuous variables and categorical variables and things like that.

Sam:
Is this covariance Matrix? Is this something that you're learning as part of the process? It's not something that you know a priori

Guest:
exactly so you can set it by Ferrari but you you can also learn as you go so there are tunable parameters around these covariance, and so it's yeah, it's Turtles all the way down. But the idea here is once you can analytically Define this is maybe a surrogate function amazing amazing option process. Here's a specific class of covariance kernels an ARD Colonel or something like that. Then you can explicitly say, okay. How good is the fit given what I've observed so far and because you're defining the system analytically need effectively mapped. The problem from is extremely sparse time-consuming and expensive underlying system that you're sampling and I wrapped it over to the surrogate space you can start to throw kind of

Sam:
The

Guest:
kitchen sink of mathematics at the problem and use that to kind of optimize the underlying various kernel pick the correct ones find the right surrogate functions and then ultimately leverage that information to decide which the the point that has the highest probability of improvement or expected Improvement or whatever. It may be

Sam:
so is the the surrogate space in this case the covariance colonel or the kind of the Spectre of this infinite Vector of the distributions

Guest:
look for various kernel defines that incident actor. Or that functional distribution. So there's two ways to think about caption process. So

Sam:
your covariance kernel is infinite by infinite Dimensions or something on that order or I mean it can how do you is part of the goal to kind of constrain the dimensionality of this Coborn's Colonel the conventional

Guest:
itself will take in inputs in the configuration space and basically say how much covariance can I expect between these two points? So it does happen to a real number technically for various types of covariance kernels. There are these tunable parameters that are continuous so like technically yes, there's an incident number of different ways you can parametrized that but we're able to do is say given what we've observed so far. What's the most likely parametrization or what the distribution of likely privatization and leverage that to decide? Okay. This is what we think is a reasonable surrogate function and then once again do that Fossil wide

variety of

Sam:
okay. I'm still not fully getting the where the infinite distributions come in. Yeah.

Guest:
So there's two ways to think about adoption process one is from the point-wise perspective. And so the idea is that every single point we're going to assume the response from this underlying system that we're sampling is going to be auctioned distribution, but every single potential configuration has a different potential couch in response to it. So

Sam:
there's some mean anything so you've got an input point and then you've got the space of configurations and each of those considerations. Translate this input point to a different distribution. The input point is a potential

Guest:
configuration. So what maybe I'll take a step back and do it except example there. So let's say we're tuning some neural network and we want to find the optimal learning right? So maybe initially we try something like point five or something like that. We get a response back. Okay, and we're optimizing for the accuracy of of fraud detection Pipeline. And so we'd be like, okay we get point seven cross validated AC that's how it looks. All right.

Sam:
So the thing that we're optimizing for is are learning rate and input is, you know, we're not talking about inputs to our neural network and help us or Internet. We're talking about an aggregate the

Guest:
well, so the inputs are we're going to be tuning this machine learning pipeline. And so at this high like meta optimization later if we're going to be saying, okay, we're going to put in a learning right and then we're going to go through the training and cross-validation and all sorts of things and come up with some Metric that we care about. So maybe cross validated AZ, right? And our goal is to find the learning rate that Tunes this entire pipeline in such a way that it maximizes that outlet. And so the way that this works in the sequential model-based optimization framework is okay. So we sampled point five learning right got point seven out as as the result and maybe there's a little bit of uncertainty associated with that. So then let's say we want a model what we think is going to happen if we try point six so we have a little bit of information because we've already sampled point five but we do is we build up this couch and process that says, okay. I'm pretty sure that it's going to pass near this point that I've already sampled, but then maybe the information dictates pretty rapidly. So I expect to see maybe point six plus or minus point one if I were to sample a point further away from And when you can think of it every potential input learning right to tune. This pipeline has its own couch and responsive were expecting it has its own mean it has its own variance and so we can explicitly build that up. Once we Define the covariance kernel and of course as you expand this out into more Dimension, so in this example,

Sam:
we're talking about what is the covariance going to look like? Yeah, so we would

Guest:
explicitly set a covariance kernel like an ARD Colonel that says okay, we're expecting some sort of like squared exponential decay of this information from sampling these different points

Sam:
and so is the covariance colonel again in this particular case. It's going to be it's going to describe the relationship between the learning rate and the output so it's going to describe the relationship between

Guest:
like individual samples of that learning, right? So does that vary where we expect wildly different results after .01 increments or is it .1 instruments? Do we expect to be an extremely noisy response or do we expect it to be fairly? Well behaved there's various different parameters of this covariance Colonel that basically say how much information effectively do I get after sampling point a about some other point

Sam:
B is the dimensionality of the covariance colonel. 6 when we start or does it increase in dimensionality as we sample,

Guest:
so it takes in the input which is the actual configurations. So in this case, it would just be a one-dimensional just the learning right but you can imagine us extending this out. So it takes an effector which is a specific configuration or two vectors actually and says, okay how much cool variances there between these two points to potential configurations. That being said you can parameterize that covariance colonel in different ways depending on which specific currently effect. So when something like an ARD Colonel, which is this this squared exponential drop off those various links skills that you can do. So

Sam:
maybe we know I don't even really drop off is that can yeah, does it

Guest:
vary over point one? But then something like the number of hidden layers might vary over orders of magnitude larger. So like 100 and layers is very similar to 101 but very different than 200.

Sam:
I'm still not sure that I'm Very clear on the the colonel and this specific example, right the dimensionality of the colonel is one by one like

Guest:
if there isn't a single value, so that's just the learning right right. Well, so do you

Sam:
think of it as a I'm thinking of it as a a matrix? Is it a function or is it something up? Should I not be thinking of it makes you can Define it as

Guest:
a matrix or it's every point the the pairwise covariance of every point you've sampled so

Sam:
far, right? So as you sample, the dimensionality of this thing is growing

Guest:
of the underlined covariance Matrix, but the underlying covariance function is just a function. So there's no

kind of dimensionality associated with it.

Sam:
Okay, so

Guest:
it's basically if I've sampled ten different points, then I could have it ten by ten Matrix, which is the covariance Matrix where every single actual instance inside that Matrix is how does point seven cool berry with X-ray or whatever it may be and this has a whole helps us Define negation process which then gives us this this acoustic surrogate function for what we think is going to happen. If we sample outside of the point that we've already explicitly

Sam:
observed. Okay, and it does that by way of defining the colonel. So how do we get from the kernel some of the Matrix to the colonel? Is that the other way around so you start

Guest:
with a colonel and then the colonel Define The Matrix though every single individual value within that Matrix is defined as I got

Sam:
it. So we're specifying the colonel in this case. She said a t r n a r d. So the what is Ard stand for? I meant working on that.

Guest:
What is the

Sam:
square Gallatin, So what's now unclear from me is if you've picked a sample in your input space and you run your your underline process and you have an output value from that Temple is the convenience Colonel used to build up like what you expected to see and then you push that all through and you get what you actually saw and

Guest:
then you can update the covariance colonel and then that covariance Matrix gets one more row and one more column because now we have how this new Point varies with all of the previously observed points and then we can use that to update our adoption process and then we have the new posterior result that we can

Sam:
use to

Guest:
decide what we sample next. And what we're doing is we're not just kind of doing naive optimization on that couch and process response itself. We don't just want to find the point with highest mean Or something like that. What we want to do is apply and acquisition function to it and say given this is what I think is going to happen. If I sample any of these potential input points, how do I find the point with the highest expected Improvement for the highest probability of improvement or which one's going to give me the most knowledge about the eventual Optimas the knowledge radiant method

Sam:
and so acquisition function is the new term that you just introduce. Is that something that is model based like the covariance kernels model based on the city or do you pick a a model that you use for your acquisition function as well?

Guest:
Yeah. So this is the optimization part of so the sequential part of sequential model-based optimization is leveraging the history to build up these sort of the models of covariance, you know this and that keeping it updated and all that stuff. The model based part is actually deciding. Okay. This is what we think the response is going to be on sampled configurations. So that's the ocean. Process in the optimization component is given that sort of model. What do we actually optimize for sampling next before we repeat this entire process? And

Sam:
so that particular piece is really focused on you know, you've got this massive potential space space for your hyperparameters. You know, how do we how do we choose a sample path through the hyperparameter space that minimizes basically wasting time and not adding information to

Guest:
exactly and this is what really controls that Explorer exploit trade-off. So a popular acquisition function is expected improved and that is basically how much do I think I'm going to beat the best thing I've seen so far by so if I've seen a pretty good AC in my front detection pipeline now, all of a sudden I want to be able to do is as well as possible beyond that we're playing King of the Hill effectively another popular one. That's kind of maybe a little bit more intuitive to it grasp is Probability of improvement if I were to sample this unsampled point what's the probability that I beat the best thing I've seen so far and so these have different exploration exploitation trade-offs and so far as probability of improvement might be a little bit more conservative. Like we're going to kind of keep edging it up slowly or is expected Improvement kind of takes the magnitude of the game into your account. So it might try something far away because I think there could be something great that it has just never seen before.

Sam:
Yeah. Yeah and are there other common examples? Yeah.

Guest:
So another one Unfortunately they get a little bit more complicated to internalize but another popular one is not as great as what my PhD advisor worked on during his PhD on the ideas.

Sam:
I'm imagining from the name like that's kind of based on information Theory and like how much we're going to learn by checking this point exactly. And the

Guest:
goal is to learn as much as we can about that eventual best point, right? So it's there's more information theoretic acquisition function and then you can kind of Define any You want with a goal of eventually getting to this best once these are probably the three most popular but you can imagine doing composite events or some sort of like upper confidence bound base acquisition function. And the idea is you wanted to efficiently as possible trade off exploration and exploitation because learning about that underlying system and how it performs and things like that's important. But at the end of the day you just want the best performing model. Yeah.

Sam:
Yeah, I think Turtles all the way down. The strikes me is that like it's you've got you've got high performance for your model. You've got hyperparameters for your pipeline and then you've got hyperparameters for your optimization system. Yeah, and presumably I'm imagining that you are also trying to optimize I prefer more High performers at the top layer for your optimization. Yes them as well. And this

Guest:
is exactly why Steakout exists because there's some incredible research out there a lot of numbers of our team contributed to the economic research and one of the open-source out there. There's a lot of promise that they should optimization have but unfortunately a lot of expert time is wasted optimizing the optimizer figuring out the best way to tune all of these Turtles all the way down. And I think that's one of the places where at least the open source that I released the network optimization engine, even though it was very popular on GitHub and it kind of failed to deliver on that promise because it required an expert to sit and fine-tune all these different things. So the goal of a company like stick out there. Can we optimize the optimizer for you and create this automatic Ensemble that makes all of these trade-offs so that you as an expert can focus on fraud detection and we'll focus on black box optimization

Sam:
for you. Okay? And so, you know, we've described the bunch of different kind of variance in this process Are there specific, you know in variance for pickup in your process like, you know, like for example, you know basing everything on a Bayesian process. That's one way of doing this like it is the product based around that and and what other kind of invariants are there in the way you approach this. Yeah.

Guest:
So at the very highest level we're just black box optimization. So there's inputs to assist them. There's an output or set of outputs that we want to optimize and we're going to try to come up with the best set of inputs. So patient optimization is an extremely efficient way to do this, especially when it's time-consuming and expensive to sample that underlying system. There's lots of different variants of Bayesian optimization. So instead of using like a option process we can use the bay General Network for the underlined surrogate function instead of using Bayesian optimization. We can use genetic algorithm particles. Simulated annealing even just a convex gradient based method the idea of being sick of takes care of that that optimization of the optimizer and automatically selects the best one for you. Most of our methods that were almost all of our methods our vision and nature but we're not constrained to

Sam:
that necessarily. Yeah, I guess that was the question that I was trying to get at. Like do you how far do you go do you you know also now or Envision a future where because you're providing the Black Box capability, you know, you may you know, do you know the Bayesian optimization but also, you know sample or test, you know, the results that you get from parvo forms and other types of methods

Guest:
definitely so in house, we built a very robust evaluation framework for deciding whether or not specific out with them farewell in different contexts. This is what we use when we integrate a new paper and want to make sure that Hi statistical confidence. It actually outperforms we're currently doing and we use this as kind of our our internal metric for deciding what to do, but we're agnostic to the end alignment. We just want the best possible thing for our customers. It turns out for the types of problems that were attacking Vision optimization is an incredibly good fit and it's kind of underutilized because it's so difficult to get up and running and and optimized but we have and will continue to employ whatever the best method is for the problems that were talking and because we Define this barrier in this way or it's just black box optimization. The underlying

system is a black box to us, but we're also a black box to our customers. And so this allows us to kind of hot swap in the best possible technique to solve their problem and not be constrained in that way.

Sam:
Okay? Okay, cool. Can you talk a little bit about the model evaluation framework that you built? Yeah.

Guest:
So there's some ice email works out papers from 2016 that go into quite a bit more detail available on our website. But the idea is I'm just told you that we have an optimization from right they can solve any kind of underline Black Box function like the First Response should be how do I know whether or not it's working so internal we built up the system. We're kind of traditionally to to publish papers and I'm guilty of doing this is you would come up with some strategy take three to six of your favorite functions show that you can outperform some specific techniques on those functions publish a paper rinse-and-repeat. So when we built this up internally, we took the super set of all of those different functions in the academic literature, which it functions that looks similar to our customers data, which took a bunch of open machine learning data sets and strategies. We basically piled them altogether. So instead of comparing against three or four different response surfaces. Now, we're looking at hundreds or thousands of them in addition to that. We wanted to make sure against all of these different open source methods and against all these other kind of different Global optimization strategies that we could very robustly outperform them. So what we do an internal evaluation framework is we independently optimize these hundreds of different pathological and real-world problems many times with cigarettes and many times with another method and that other method might be just a new version of cigarettes and then with high statistical confidence we can say which one got to the best value fastest which one got to the ultimate best results, which one was the most robust. So it didn't have like in the interquartile ranges are all above a specific value.

Sam:
It sounds like to draw an analogy from software engineering building regression testing framework for Optimizer.

Guest:
Yes. So we do use it for regression testing its run nightly, but it's also a way to basically a B test optimizers. Right?

Sam:
Right. You're not using it to or what extent are you using it to inform model choices or I guess that you know what I'm struggling a little bit with is, you know, so you've got the You've got this, you know, this heat of data sets and functions and things like that. And if you were trying to optimize across all of those then you've got at least common denominator kind of problem, right or you know, local Maxima or something like that. Yeah.

Guest:
So we do have to be wary that. We don't over fit to the state of side. That's definitely true. One thing that we found though is the reason why we built an ensemble based

Sam:
approach. So let me just just poke it that like I'm not sure is over putting the right word for what I'm thinking of is is that you know, some of the strikes me as the opposite of overfitting. Whereas like if I were to just look at I don't really care about all this other date I care about my problem like if you're optimizing for this kind of broad spectrum and I can you know outperformed you by just focusing on my problem, you know, I'd probably do that. Yeah

Guest:
that makes complete sense. I see where you're coming out here. So this is why we take this Ensemble based approach because it turns out like the most popular approach to invasion. Optimization like the ocean processes with arid Eternal with expected Improvement actually doesn't do super well and a wide variety of different contexts. So by splitting in the right tool for the job we can actually hit all of these different facets of different types of problems extremely. Well, that being said no free lunch there. I'm in computer science still applies here and so far as if you do have expert knowledge about your underlying system and you build with the spoke Optimizer to solve that one specific problem. You are going to outperform a general technique that being said, You would have to repeat that for the next problem with you attack and the next one of our next month. And so the idea is by having an ensemble of different optimizers. We use the right one for specific context and then a different one for a different context etcetera. So instead of having like the lowest common denominator, like you said just the one-size-fits-all we're doing is actually putting in the right tool and automatically learning when we traded off. So when you're tuning a gradient boosted method you're getting the right tool, but when you to neural network, it's still the same API instead of interface, but you're getting the right optimize right?

Sam:
So what I'm hearing is in response to my question like a little bit of go, right like you're you built this model evaluation framework because fundamentally, you're not necessarily trying, you know, outperformed handcrafted Model 50 PCS has been five years developing whatever you're trying to build a system that can deliver good performance on you know in general what someone Throws at it. And so you want to test it against a bunch of you know, these are things that someone might throw at it and make sure that you get good performance. And the way that you do that is under the covers you not just relying on, you know, one specific set of choices, but you're taking an ensemble approach and your Optimizer can swap in and out different decision to produce a result that that's about

Guest:
that's exactly because what we find more often than not is that people don't assign fifty phds for five years for every single optimization program. They have more often than not they could search random search manual tuning Navy and open source solution. Maybe they have part of their team part-time working on an internal Optimizer or something like that. And those are the things that we can vastly outperform if you know, it's contacts and you know grading information and you have a bunch of expert knowledge like there is specific tools that you can use to get there and this is probably a little heavy-handed. Teams in that situation but more often than not what we're doing is we're coming in replacing these very exhaustive very expensive very domain expert intensive systems and we can generally outperform those to a high degree,

Sam:
you know, and like to think of the the tools base in general is like there's you know for many Enterprises, there's such a huge potential opportunity to apply ml that their ability to staff up, you know as far outpaced by the opportunity so they given Staffing level like you've got this choice you can either like, you know, take only the the biggest opportunity and apply all your resources to that in a very manual way or you can you know, utilize tools that allow trucks to be more effective and fight off some of these, you know, some some of the it just like the a lot of and I'll talk to folks in to talk about it. Like we only go after home runs vs. You know base hits right and This is sounds like this is a tool for long people to you know, won't both go after home runs as well as try to increase their hit rate for bases.

Guest:
Definitely what we find with a lot of the firms that we work with is how they differentiate themselves from the

competitors is not by Black Box station optimization. It's like creating a great recommendation engine or wait out like trading started and if you can hire five more phds to to work on that core differentiator or free up five phds to do that and then just use the opportunity. They work very additively hand-in-hand. We can accelerate that time to Market accelerate the results getting to the best performance and all of these different things and I think more and more companies are becoming aware of this and using the right tool for the job why we write tensorflow when you can use it why right around Beijing Optimizer when you can use a best-in-class easy recipe awesome awesome.

Sam:
So what's the what's the best way for folks to learn more? I'm assuming the website

Guest:
or Miss contact that's it. Dot-com. If you want to shoot us an email run a complimentary proof-of-concept pilot. Like we can throw these peer review papers that you to prove that we're as good as we say we are but at the end of the day we want to prove it with their underline models themselves so we can work with any Enterprise any underlying system Cloud agnostic model agnostic. It's also free for students. So if there any people at universities or researchers the National Labs or whatever it is listening to the podcast pickup.com / heating unit free enterprise account. I wasted way too much my parents be on the problem don't want to do that for anybody else. And

Sam:
what about for folks that are interested in learning about the theoretical foundations of the work. Where would you point them or their life and three canonical papers or something like that that they should look for? Yeah.

Guest:
So if you go to stick up. Com slash research, those all of our papers. We also have a Bayesian optimization primer there that kind of goes into more detail about some of the things I said verbally sometimes a little bit hard to describe the ocean processes and things like that. And that is there. There's references for all those papers as well so I can kind Take you down the rabbit hole of all the different ways that this has been applied the store.

Sam:
Okay. Awesome. Well, thanks so much Scott. It's been a great conversation, and I've learned a

Guest:
ton. Excellent. Thank you so much. I really appreciate it. Thanks.

Sam:
All right, everyone. That's our show for today. Thank you so much for listening. And of course for your continued feedback and support for more information on Scott and the topics covered in this episode head on over to twimlai.com slash talk slash fifty next week on Tuesday and Wednesday, October 3rd, and 4th. I'll be at the Gartner Symposium in Orlando where I'll be on a panel on how to get started with a i if you'd like to meet up there. Please send me a shout.

Guest:
The

(end of excerpt)

Conclusion

Sam:
following week. I'll be in Montreal for the rework deep learning Summit and help to be joined by at least one lucky listener. Remember to visit twimlai.com slash DL Summit to enter contests and the new Central on October 4th. Thanks again for listening and catch you next time.

(end of excerpt)