EPISODE 95

[INTRODUCTION]

[0:00:10.8] SC: Hello and welcome to another episode of TWiML Talk, the podcast where I interview interesting people doing interesting things in machine learning and artificial intelligence. I'm your host, Sam Charrington.

Just a couple of quick announcements today related to the TWiML online meetup. First, the video from our December meetup has been posted and it's now available on our YouTube channel and at twimlai.com/meetup. It was a great meetup. If you missed it, you'll definitely want to check it out.

You definitely don't want to miss our next meetup either. On Tuesday, January 16th at 3:00 Pacific, we'll be joined by Microsoft researcher, Timnit Gebru, who will be presenting her paper Using Deep Learning and Google Street View to Estimate the Demographic Makeup of Neighborhoods Across the United States, which has received national media attention for some of its findings. Timnit will be digging into those results, as well as the pipelines she used to identify 22 million cars and a 50 million Google street view images.

I'm anticipating a very lively discussion segment as well to kick off the session. Make sure to bring your AI resolutions and predictions for 2018. For links to the paper or to join the meetup group, visit twimlai.com/meetup.

All right, now a bit about today's show. In this episode, we hear from Siddha Ganju, Data Scientist at Computer Vision Startup, Deep Vision. Siddha joined me at the AI Conference a while back to chat about the challenge of developing deep learning applications "at the edge." In other words, those targeting compute and power constrained environments.

In our conversation, Siddha provides an overview of Deep Vision's embedded processor, which is optimized for ultra-low power requirements, and we dig into the data processing pipeline and network architecture process that she uses to support sophisticated models and embedded devices. We dig into the specific hardware and software capabilities and restrictions typical of edge devices and how she utilizes techniques like model pruning and compression to create embedded models that deliver needed performance levels in those resource-constrained environments.

We also discussed use cases such as facial recognition, scene description and activity recognition. Siddha's research interest also include natural language processing and visual question answering. We spent some time discussing those as well.

Now, on to the show.

[INTERVIEW]

[0:02:49.7] SC: All right, everyone. I am here at the Artificial Intelligence Conference in San Francisco and I'm with Siddha Ganju, who is a Data Scientist at Deep Vision. Siddha, welcome to the show. It's a pleasure to have you.

[0:03:03.2] SD: Hi. Thank you very much. Thanks for having me.

[0:03:06.2] SC: Absolutely. Absolutely. Why don't we start by having you tell us a little bit about your background and how you got interested and started in machine learning.

[0:03:16.4] SD: I think I got started in machine learning during my undergrad days. I had gone to a Hackathon and I met this mentor there. His name is Anirudh Koul. We worked on a project there, which was called orphan locator, which was basically trying to locate missing children using the police databases.

We used a very simple image matching algorithm there. I think that was my first introduction to machine learning. Then after the Hackathon when I came back to college, I was like, "I want to know more about." I think like everybody else, I started doing the course over course on machine learning.

[0:03:53.7] SC: The enduring course.

[0:03:54.4] SD: Yeah. I guess, after that I applied for a master's degree in data science. I just graduated this year from Carnegie Mellon with a Masters in data science. Then at CMU, I worked on what is called visual question answering. That's an AI hard task, which basically provides an image to a computer and user, or human as expected to ask a question about the image.

Now this question can be about any activity or the number of people, or something related to the image or the scene within the image. The computer or the AI system is expected to provide an accurate answer to that question. Now there are many uses of Milky Way of visual question answering. One is obviously for the visually impaired.

Another use is for people in situationally impaired, for example if you're in a car and you're driving. You don't want to be looking at your phone. Your phone can basically give you a description of the images that somebody just sent to you, or if you're a security analyst then you don't have to comb through hours of video footage. You can just query, like what did the man take from that shopping mall. You can just describe the situation sort of to a system, and the system can provide you those frames in which that happened. These are some of the examples of Milky Way.

Our research was focused on how can we use visual questions as a form of supervision for improving computer vision models? Because in the future, it will become common for humans to ask visual questions to computers like, "Where did I leave my keys? Or what breed of dog is this?"

Now, if you look at this question there is a lot of information already provided in the question itself, like the object or the animal we're talking about is a dog and etc. That's primarily where the research was focused.

[0:06:06.5] SC: Okay, interesting. Here at the conference, you did a talk on your – I forget you're doing, or you did it yesterday.

[0:06:14.6] SD: I did it yesterday.

[0:06:15.8] SC: Right, right, right. That talk wasn't on VQA. That talk was on –

[0:06:19.5] SD: Yeah, that was actually on embedded deep learning, which says how you can take deep learning algorithms, which are compute intensive. They are pretty big in size and how you can take them to embedded devices, because embedded devices have limited compute available and they have limited storage. How can you run these algorithms at inference time on these devices? This is basically the work that I'm doing currently at my company, Deep Vision.

[0:06:47.8] SC: Okay. Is that the focus at Deep Vision, or is that just one of the many things that the company is working on?

[0:06:54.7] SD: That is the focus of Deep Vision basically to –

[0:06:58.4] SC: Tell us a little bit about the company.

[0:07:00.2] SD: Yeah, sure. The company was founded by two Stanford PhD graduates, Rehan and Wajahat. The hardware or the processor that they developed was during their PhD itself at Stanford. This hardware is basically – it has high performance forward; at the same time, it is programmable enough, so you can run a wide range of algorithms, which includes both traditional computer vision algorithms and deep learning algorithms on the same device itself.

If you look at most of the processor these days, if you want high performance then it's ideal to develop what is called a custom hardware, or a fixed function hardware, which is built basically for that one particular operation that you want.

On the other hand, if you want a broad spectrum device, which is programmable so you can run a lot of things on it, it will be not as efficient as the fixed function hardware. An example of programmable devices are the GPUs, or the graphical processing units. But these have high costs, so they're expensive. They're also really big, so you can't actually put them on embedded devices. They were able to figure out a way to bridge the gap between performance and programmability through which they developed this processor.

[0:08:36.8] SC: Okay. Interesting. Does the company compete with, or play in the same space as the Intel Movidius?

[0:08:45.8] SD: It's actually a little different, because we are building both hardware and the software. I don't think Movidius follows that plan. Yeah.

[0:08:57.0] SC: Okay. The hardware is specifically focused, I'm inferring from the name Deep Vision, on visual types of problems and CNNs for example?

[0:09:10.6] SD: We can run CNNs. We can also run LSTMs on it. It's not particularly just the convolutions. Yeah, a broad range of deep learning basic systems can be run on it.

[0:09:23.6] SC: Why don't you walk us through your talk and the major points that you were trying to convey to the audience there.

[0:09:32.1] SD: Sure. As I already mentioned about the hardware innovation, which is bridging the gap between the performance and the programmability. One of the basic ideas behind this is that the convolution operation, that basically belongs to one of the classes of those computations for which it's possible to build efficient hardware when you will let in ASIC format, or Application Specific Integrated Circuits. That's basically what they did.

They were able to basically optimize this convolution. Now if we look at traditional computer vision methods, most of them have a overlapping stencil or like a sliding window on which they run operations, which if you think about it is similar to a convolution.

Additionally, it's also similar in like MapReduce operations. That's also like a window and you're repeating the method over and over again over different windows. This is basically how they got the idea to optimize this one particular class of functions, and it was wide applicability or traditional computer vision and deep learning algorithms.

The other thing that I mentioned in the talk was that why are we focusing on embedded devices, or edge devices. If you look at the data available for embedded devices, it's approximately more than 150 Zettabytes of video data.

If you think about where this is coming from, like airport surveillance cameras, traffic light cameras, basically all the cameras that you see anywhere, they have embedded devices in them. They need someone to be looking at the videos right now. It is a hope that these can be automated eventually. That's where you will need these devices to be intelligent enough to perform real-time analysis.

[0:11:36.9] SC: The idea is that you've got tons and increasing amounts of surveillance data all over from security devices, smart city initiatives –

[0:11:47.2] SD: Even a home security.

[0:11:49.3] SC: - down the home security and eventually maybe our phone camera also will be always on.

[0:11:54.9] SD: I don't know. Yeah.

[0:11:56.6] SC: There is some people word about scenarios like that. But in any case, there is just tons and tons of video data all over the place. Right now, people are reviewing that manually and you would – the company is building towards a model where you're training models to identify various features, like objects or people, or things like that and you would deploy those models out to devices, inference engines that live at the edge, and can basically raise flags when different things are happening.

[0:12:33.1] SD: Yeah. One more thing is that the models that you train, like for home security it will be different than for example airport security. Because in home systems, you need to recognize five or six people; not more than that. An airport, you need to recognize thousands of people instantaneously. The way of training the models and developing the models in both the scenarios is completely different.

We're also looking into how to train each one and how to make each one dense enough, so that the model is extremely small, so that we can fit it on to these embedded devices. At the same time, they should be accurate enough that we are getting the correct results.

[0:13:17.6] SC: You mentioned the number of people that you're trying to identify. It sounds like one of the main use cases is in the security scenario I see this person in this frame here pull up other frames in videos where that person appears. The salient point being not just identifying when there are people, but it's identifying specific people. Maybe what are the specific use cases or that tied a specific model classes, I guess?

[0:13:54.4] SD: I think once the technology is in place, the possible use cases are endless. Right now, we're focusing on two main ones. Again, I talked about both of these yesterday. These are face recognition and scene description. Face recognition is basically finding out the name of a particular person based on the image of that particular person. Scene description is giving out a caption, or a description for a scene, which is within the image.

Now, scene description has uses in home security systems. Because right now the home security systems as such that they alert you that there is motion detected outside your house, or something is happening, but they don't tell you what is happening. Sometimes these –

[0:14:43.7] SC: For example, the UPS is here with the package.

[0:14:46.1] SD: Yeah. Our system can say, "Okay, the UPS guy is here with a package and it's dropped on your front door," or like similar things like that. Again, face recognition can say if – say you saved your son or your daughter's face in your system, you'll know that they're coming home. The system just tells you, "Okay, this person has reached home." That's another use case.

[0:15:11.3] SC: Yeah, it's interesting. I think, at least for me as you were describing what you were trying to do, it's easy to get carried away. Imagine tons of use cases. When you think through the kinds of models that they would require, they're all really different. You have to really focus, at least now on some very specific use cases.

[0:15:33.1] SD: Yes, that's true. Another thing is that in order to develop say even activity recognition, you need to have some basic recognition capabilities. For example, you need to define what an arm is, or any other body part. That recognition capability comes from what is possible again in face recognition. What I'm saying –

[0:15:58.3] SC: Okay. Elaborate on that.

[0:15:59.3] SD: Yeah. It's basically when you have one face recognition model in place, the only thing you have to do is tweak it a little bit to make it into activity recognition. Like in place of images and face recognition, you would need a time sequence, or a frame sequence and activity recognition.

[0:16:22.4] SC: Are you talking here about transfer learning? Meaning, you've trained the model on faces and now you can use it to identify arms? Or are we talking more about sequence-related things, or something totally different?

[0:16:35.5] SD: I think it's something totally different.

[0:16:36.4] SC: Okay. Great.

[0:16:38.4] SD: What I mean is that once you have the basic capability in place, like being able to recognize faces, it is just parts of these algorithm or parts of this model that you would be reusing in other models, like activity recognition.

Now you won't be using the exact same weights, because that would be completely different. You would have to retrain the – Or actually not retrain. Train from scratch the active data recognition model. That's at the basic elements in both of other thing, like the convolutions or the LSTMs. Yeah.

[0:17:14.1] SC: Okay. You're saying you're building up the model architecture share a lot of common characteristics.

[0:17:21.1] SD: Yeah. Something like that. Yeah.

[0:17:22.7] SC: Okay. Something like that, but not quite that.

[0:17:24.5] SD: I didn't actually understand what you said.

[0:17:26.1] SC: The model architecture share a lot of common characteristics, meaning you're using the same general model architectures; the different types of layers, convolutional layers and –

[0:17:36.0] SD: Yes, yes, yes. That's very easy way to explain it. Yeah. I should've said that.

[0:17:40.9] SC: Interesting. I guess, what I'm curious about is you develop models – I'm assuming that the way that you would go about this is you want to develop facial recognition models and you surveyed the literature, figure out what are the best performing model architectures to do that, implement those, train those. Then you've got this model that probably doesn't fit on your embedded device. There is a process that you go through to go from that model to one that fits. Walk us through how much of that is art and how much of that is science and walk us through the thinking as you do that.

[0:18:29.7] SD: Yeah, sure. This process is actually called pruning. How you can go from a big model to something is just 11 – reduced to 11 times its original size. The way to do this is – Pruning basically has three different steps in it. Well, I think there are three different steps in it. First is you need to statistically analyze your model to ensure that the weights follow a bell curve distribution, like a normal distribution.

[0:19:01.6] SC: Yeah, I get that part. I think you're responding to me looking up like – Okay, why is it important that your weights are distributed in that way?

[0:19:08.6] SD: Yeah. Because when you're going to prune it, you're going to use some thresholds. Now you calculate these thresholds using the standard deviation. The assumptions of standard deviation are that it needs to be a bell curve.

[0:19:25.8] SC: Is it typical, or common that your weights do follow the bell curve?

[0:19:30.6] SD: It's common. Yeah, for all the models that I've tried it with, they almost always fall into a bell curve. Once you know that it's a bell curve, you move on to the next step which is the actual pruning stage. You calculate the standard deviation off the weight matrices, then you find the quartiles of each weight matrix. That's basically standard deviation multiplied by one, two, three, four and so on.

Now for each weight matrix, for example if you're using a scene description model that will have an image model and a language model. For each of these, you will calculate their thresholds. Then you can remove all the weights, which are less than that threshold. The first threshold you calculated say it was zero. Any number less than zero, you can remove it.

[0:20:27.8] SC: Okay. This is essentially a technique to identify and rank the contribution of the visual weights in your model.

[0:20:37.3] SD: Yes. Yes. Yeah.

[0:20:41.1] SC: Okay. Then you rank order these weights in terms of their contribution and you have some cutoff and you just remove the weights that are – that fall beneath that cutoff. Now, I'm imagining when you do that, there are ripple effects in terms of your –

[0:20:55.9] SD: Yeah. That's why you need to load these weights back into the model and then retrain it.

[0:21:03.5] SC: All right. Was that your third step?

[0:21:06.0] SD: Yes. Retraining is the third step. Yeah.

[0:21:08.1] SC: Awesome. Awesome.

[0:21:09.3] SD: Then you can basically repeat this entire process until you reach the most dense model.

[0:21:16.8] SC: Okay. Is there empirical work that shows that pruning leads to optimal compact solutions relative to some other process that maybe starts from a smaller or more compact model and trains those from scratch or something?

[0:21:37.5] SD: There are actually different kinds of pruning strategies available. I remember, there are a couple of papers from Stanford that talk about this method. There are a couple of papers from University of Washington and Allen Institute that talk about just removing one complete branch; zeroing out everything in one convolution, and then retraining it. It really depends on what kind of model you have and what results you want to attain. Yeah.

[0:22:13.0] SC: All right, interesting. Can you give us a sense for the kinds of results, well you mentioned that your models after the pruning process can be 10% of the size of the original models. In real numbers, what does that look like?

[0:22:30.6] SD: Sure. If you talk about the scene description model that we worked on, its average inference time, it came down from 8 milliseconds to 2 milliseconds. The accuracy also increased – for scene description that are different metrics, like METEOR, BLEU, ROUGE and CIDEr. There was an increase of approximately five steps on these metrics.

[0:22:55.9] SC: Say that again, METEOR, BLEU, ROUGE and –

[0:22:58.3] SD: CIDEr, yeah.

[0:22:59.0] SC: CIDEr?

[0:22:59.8] SD: C-I-D-E-R.

[0:23:00.7] SC: Okay.

[0:23:02.5] SD: B-L-E-U. These are some image captioning metrics. These are originally machine translation metrics, but they have been adopted to image captioning metrics. There is also a new metric called SPICE, which is used for image captioning.

[0:23:19.7] SC: Okay. You're saying that you can train a model, measure it against these metrics, prune the model and then increase performance –

[0:23:30.6] SD: Yeah, because you will have a dense network.

[0:23:31.1] SC: - relative to the expanded model?

[0:23:34.3] SD: You can change some things in the network, like change the image model to something smaller and retrain it. Because you're starting from trained weights, so you have a good initialization in your system when you retrain it. That basically helps in going above the previously attained accuracy level.

[0:23:56.8] SC: Just to make sure I'm understanding the previously attained accuracy level for an unconstrained by size model?

[0:24:05.7] SD: Yeah.

[0:24:06.9] SC: It's just it's counter-intuitive to me. The way I envisioned this is that the best performance you're going to get is when you've got a model that's not constrained by memory power, etc., then you prune it and you make some compromises and you get adequate performance, but with a model with a footprint that can fit on your embedded device. What I hear you saying is that you can actually increase your performance and shrink your model down at the same time? Is that what you're saying?

[0:24:39.3] SD: Yes. Yeah. As an example of, again the scene description model; so if you start from something like neural talk that has a VTT network for its image model and a pre-trained LSTM for its language model. Now if you remove this VTT network and replace it with something smaller like GoogLeNet – GoogLeNet and VTT Net both lie within the same top 1% accuracy range. If you use GoogLeNet and the same pre-trained LSTM and retrain this entire system, you can actually get a higher accuracy.

[0:25:18.5] SC: For folks that are doing research in this area and are competing on accuracy, why don't they all just add another step in their process of pruning to come up with a better performing model, or at least try that?

[0:25:34.4] SD: Because I think that's not their main aim. Accuracy is their main aim, but pruning is not their main aim.

[0:25:42.4] SC: Right. But you're saying accuracy can improve because you pruned?

[0:25:47.0] SD: Yes, that's true. I know that because I tried that out as an experiment. I mean, if you're a PhD student, I doubt you will have time to experiment with pruning just for fun.

[0:26:00.7] SC: Okay. So that I'm not making assumptions here, are you asserting that – Maybe I am jumping to conclusions and you're not asserting that increased performance is, or accuracy is a general result as opposed to you just happened to see this one time.

[0:26:18.6] SD: It's not a general – Yeah, that's true. Yeah.

[0:26:20.8] SC: Okay. I think that's the link that I was making.

[0:26:21.7] SD: It's not a general – It's not going to happen all the time. It's not going to happen all the time, but there are some cases, like in this case because the accuracy of both image models lies in the same top 1% range, that could be one possible reason why we're seeing the increase in accuracy. If I were to use some other image model, the same results might not be repeated and the accuracy might actually decrease.

[0:26:46.1] SC: Okay. That makes more sense. Okay, so were there other things that you covered in your talk? You went through your three steps.

[0:26:54.4] SD: Yeah, that's pruning. I think that's about it. I mean, there were a lot of other things, but there is not much related to what we are talking about right now.

[0:27:05.4] SC: Okay. What were the other things then?

[0:27:07.9] SD: Like for the face recognition pipeline, that is mostly two steps, like face detection and the actual recognition part. Can you improve on each one of these individually? For face detection, if you use a standard library, or a traditional computer vision system as opposed to something trained on neural networks, can you improve the accuracy, the inference time and the model size?

For face detection, we saw improvement on all these three verticals. On face recognition, we trained different models using somewhat similar architecture. Google had released a FaceNet paper which describes the NN2 architecture. We built several models around the NN2 architecture and trained with different input sizes and saw that there's different inference time, different accuracy that it attains and different model size that all of these three – sorry, all of these two parameters can change. That's something that I also mentioned in the talk.

[0:28:16.7] SC: Okay. The takeaway there is then that if you are developing a pipeline for something like facial recognition or some of these other – let's maybe generalize it to – if you're developing a pipeline generally and you want to get that pipeline to run while in an embedded environment.

[0:28:40.5] SD: Yeah, you want to be optimizing each portion of the pipeline individually.

[0:28:44.4] SC: Right. As opposed to just optimizing – being fixed on your end pipeline and optimizing that.

[0:28:50.9] SD: Yeah. But that said, it's important to – after you're optimizing each little bit of it, you need to go over a retraining step for the entire pipeline. This end-step is I think the most important step.

[0:29:09.2] SC: Okay. You don't want to skip optimizing the individual pieces, but you want to once you've done that, optimize the end piece. Is the idea that you start you optimization of the end-to-end system with better initial weights for the individual pieces.

[0:29:26.4] SD: Yeah. That's what I understand based on all the experiments that I've done.

[0:29:32.3] SC: Okay. All right. Interesting. Interesting. Can you walk us through – we talked a little bit about VQA. Can you walk us – is that something that you work on at Deep Vision as well, or is it –

[0:29:43.0] SD: I think VQA will come in eventually, because like I said scene description is something that we use right now. Eventually, you would also want people to be asking questions to the system, so that the system can give you an answer.

[0:29:55.4] SC: Okay. Can you walk us through what the current state of the art is with VQA? What are the approaches that folks are using in generally how they take on that problem?

[0:30:08.5] SD: Sure. I don't quite remember what is the state of the art now, but generally the approach is that you take an image model and you somehow interface or communicate it with a language model which takes the question as input.

When you're interfacing these two matrixes together, the result will be a single vector, which will be the answer to the question that you have. You can replace the image model with ResNet, inception, GoogLeNet, or basically anything, or like a combination of all of these. Language model usually is an LSTM, or you can also have it as a bag of woods vector, or any other representation of the text.

Now most of the work from VQA is coming from Devi Parikh and Dhruv Batra's lab. Now, they also started using reinforcement learning in this. They're just trying to give adversarial answers and questions and having the other computer or the other agent within the same environment trying to figure out which of these is incorrect and which of these is correct. That's a brief overview of what's happening in VQA.

[0:31:30.1] SC: Okay. Interesting. There are some – I don't remember the – Maybe you can remind us the name of them. There are some popular data sets that folks are using for VQA.

[0:31:39.2] SD: For VQA? Yeah, it's called the MS COCO data set.

[0:31:41.6] SC: MS COCO?

[0:31:42.8] SD: Yeah. It's released by Microsoft and it's the common object in context. Now the images in MS COCO are actually really different from ImageNet, because ImageNet focuses on one particular object and image. Whatever the object is in object, it usually occupies most of the area within the image.

In the COCO images, they're like normal scenes. Say this room for example, it doesn't have a specific object or there is no specific person in focus. It's like random – not random, but scenes which have a lot of information in them. In fact, there is also a release of the second version of the MS COCO data set that happened this year. That data set actually fixes some of the errors in – not the errors, but actually devices in the first data set.

For example, in the first data set if you had a number of question like how many apples are on the table? Most generally, the answer would be three. Or if the question is what is color is anything? Most generally, the answer would be red.

[0:32:55.1] SC: If you train on this, you develop a model that over fits on threes and –

[0:33:00.0] SD: Yeah. They actually trained like a image blind model which never saw the images, only the questions and the answers. This kind of model would just learn that if this is the question, this is the most probably answer. Even that formed considerably well.

[0:33:17.2] SC: Because of biases like that.

[0:33:18.6] SD: Yes, exactly. That's why developed the MS COCO version 2 data set.

[0:33:24.8] SC: Okay. Great. Awesome. Well, thank you so much for taking a few minutes to chat with me. I really appreciate it.

[0:33:31.4] SD: Thank you. Thank you for having me.

[0:33:33.0] SC: I didn't mention this at the intro, but we initially got connected because you listen to the podcast.

[0:33:37.4] SD: Yeah. I actually listened to Chelsea's podcast; Chelsea Finn's. That's how I really got interested. I listened to – that NLP podcast from someone. I don't remember her name, but it was pretty recent around Chelsea's podcast. I was like, "Wow, there is so many things that I don't know."

[0:33:56.8] SC: Okay. That was [inaudible 0:33:57.4].

[0:33:58.2] SD: Yeah, maybe. Yeah. It was a difficult name for me to remember. Yeah.

[0:34:03.2] SC: Awesome. Awesome. Well, thanks so much for listening and thanks very much for spending some time.

[0:34:09.2] SD: Yeah. Well, thank you first of all for having this great idea. Thank you for having me today.

[0:34:14.9] SC: Awesome.

[0:34:15.3] SD: Thank you.

[END OF INTERVIEW]

[0:34:20.5] SC: All right everyone. That's our show for today. Thanks so much for listening and for your continued feedback and support. Thanks to your support, this podcast finished the year as a top 40 technology podcast on Apple Podcasts. My producers says that one of his goals this year is to crack the top 10. To do that, we need you to head over to your podcast app, rate the show, hopefully we've earned five stars, and leave us a glowing review.

More importantly, share the podcast with your friends, family, coworkers, the Starbucks barista, your Uber driver, everyone who might be interested. Every review, rating and share goes a long way, so thanks in advance.

For more information on Siddha or any or the topics covered in this episode, head on over to twimlai.com/talk/95.

Of course, we would love to hear from you either via a comment on the show notes page or via Twitter to @samcharrington, or @twimlai, or @twimlai.

Thanks once again for listening, and catch you next time.