Geoff Hinton Transcript Part One | The Robot Brains

76-97 minutes

Geoff Hinton on The Robot Brains Season 2 Episode 22

Pieter Abbeel

Over the past ten years, AI has experienced breakthrough after breakthrough after breakthrough in computer vision, in speech recognition, in machine translation, in robotics, in medicine, in computational biology, in protein folding prediction and the list goes on and on and on. And the breakthroughs aren't showing any signs of stopping. Not to mention, these AI breakthroughs are directly driving the business of trillion dollar companies and many, many new startups. Underneath all of these breakthroughs is one single subfield of AI deep learning. So when and where did deep learning originate? And when did it become the most prominent approach? Today's guest has everything to do with this. Today's guest is arguably the single most important person in history and continues to lead the charge today. A recipient of the Turing Award, the equivalent of the Nobel Prize for computer science. Today's guest has their work cited over half a million times. That means there are half a million and counting other research papers out there that build on top of his work. Today's guest has worked on deep planning for about half a century and most of the time in relative obscurity. But that all changed in 2012, when he showed deep learning is better at image recognition than any other approaches to computer vision and by a very large margin. That result, that moment known as the ImageNet moment, changed the whole AI field. Pretty much everyone dropped what they had been doing and switched to deep learning. Former students of today's guest include Vlad Minh, who put DeepMind on the map with their first major result on learning to play Atari games and includes our season one finale, Ilya Sutskever, founder and research director of OpenAI. In fact, every single guest on our podcast has built on top of the work done by today's guest. I am, of course, talking about no one less than Geoff Hinton. Geoff, welcome to the show. So happy to have you here.

Geoff Hinton

Well, thank you very much for inviting me.

Pieter Abbeel

So glad to get to talk with you on the show here. And I'd say let's dive right in with maybe the highest level question I can ask you. What are neural nets and why should we care?

Geoff Hinton

Okay. If you already know a lot about neural nets, please forgive the simplifications. Here's how your brain works. It has lots of little processing elements called neurons. And every so often a neuron goes ping. And what makes a go ping is that it's hearing pings from other neurons and each time it hears a ping from another neuron, it adds a little weight to some store of input that he's got. And when he gets it, when it's got enough input, it goes ping. And so if you want to know how the brain works, all you need to know is how the neurons decide to adjust those weights that they have when a ping arrives. That's all you need to know. There's got to be some procedure used for adjusting weights. And if we can figure it out, we know how the brain works.

Pieter Abbeel

And that's been your quest for a long time now, figuring out how the brain might work. And what's the status? Do you, do we as a field, understand how the brain works?

Geoff Hinton

Okay. I always think we're going to crack it in the next five years since that's quite a productive thing to think. But I actually do, and I think we're going to crack it in the next five years. I think we're getting closer. I'm fairly confident now that it's not backpropagation. So all of existing AI, I think, is built on something that's quite different from what the brain is doing. At a high level, it's got to be the same. That is, you have a lot of parameters, these weights between neurons, and you adjust those parameters on the basis of lots of training examples. And that causes wonderful things to happen if you have billions of parameters. The brain's like that and deep learning is like that. The question is, how do you get the gradient for adjusting those parameters? So what you want is some measure of how well you're doing, and then you want to adjust the parameters so they improve that measure of how well you are doing. But my belief currently is that backpropagation, which is the way deep learning works at present, is quite different from what the brain is doing, the brain is getting gradients in a different way.

Pieter Abbeel

Now, that's interesting. You're the one saying that, Geoff, because you actually wrote the paper on backpropagation for training neural networks, and it's powering everything everybody is doing today. And now here you are saying, actually, it's probably time for us to figure out, well do you think we should change it to be close to what the brain is doing, or do you think maybe backpropagation could be better than what the brain is doing?

Geoff Hinton

Let me first correct you. Yes, we did write the most cited paper on backpropagation from Rumelhart, Williams and me. Backpropagation was already known to a number of different authors. What we really did was show that it could learn interesting representations. So it wasn't that we invented backpropagation. Rumelhart really invented backpropagation. We showed that it could learn interesting representations, like, for example, word embeddings. So I think backpropagation is probably much more efficient than what we have in the brain. But squeezing a lot of information into a few connections whereby few connections, I mean, only a few billion. So the problem the brain has is that connections are very cheap. We've got hundreds of trillions of them. Experience is very expensive. And so we are willing to throw lots and lots of parameters at a small amount of experiments, whereas the neural nets we're using are basically the other way around. They have lots and lots of experience and they're trying to get the information about what relates the inputs, the output into the parameters. And I think backpropagation is much more efficient at what the brain is using, doing that, but maybe not as good at, from not much data, abstracting a lot of structure.

Pieter Abbeel

And well, this begs the question, of course, do you have any hypotheses on approaches that might get better performance in that regard?

Geoff Hinton

I have a sort of general view which I've had for a long, long time, which is that we need unsupervised objective functions. So I'm talking mainly about perceptual learning. Which I think is the sort of key. If you can learn a good model of the world by looking at it, then you can base your actions on that model rather than on the raw data. And that's going to make doing the right things much easier. I'm convinced that the brain is using lots of little local objective notions. So rather than being a kind of end to end system trained to optimize one objective function. I think it's using lots of little local ones. So as an example, the kind of thing I think would make a good objective function, thought it's hard to make it work, is if you look at a small patch of an image and try and extract some representation of what you think is then some. You can now compare the representation you got from that small patch of image with a contextual bet that we've got by taking the representations of other nearby patches. And based on those predicting what that patch of the image should have in it. And obviously, once you're very familiar with the domain, those predictions from context and locally extracted features will generally agree. And you'll be very surprised when they don't. Then you can learn an awful lot on one trial if they disagree radically. So that's an example of where I think the brain could learn a lot from this local disagreement. It's hard to get that to work, but I'm convinced something like that is going to be the objective function. But if you think of a big image and lots of little local patches in the image, that means you get lots and lots of feedback, in terms of the agreement of what was attracted locally and what was predicted contextually, all over the image and many different levels of representation. And so we can get a much, much richer feedback from these agreements with contextual predictions. But making all that work is difficult. But I think it's going to be along those lines.

Pieter Abbeel

Now, what you're describing strikes me as part of what people are trying to do and self-supervised and unsupervised learning. And in fact, you wrote one of the breakthrough papers, the SimCLR paper, with a couple of collaborators, of course, in this space. But what do you think about the SimCLR work in contrast of learning more generally? And what do you think about the recent masked auto-encoders and how does that relate to what you just described?

Geoff Hinton

It relates quite closely to what, it is evidence that that kind of objective function is good. I didn't write the SimCLR paper. Ting Chen wrote the SimCLR paper. My name was on the paper for general inspiration, but I did write a paper a long time ago with Sue Becker on the idea of getting agreement between representations you got from two different patches of image. So that was, I think that was the origin of this idea of doing self-supervised learning by having agreement between representations from two patches of the same image. The method that Sue and I used didn't work very well because of a subtle thing that we didn't understand at the time. But I now do understand. And I could explain that if you like, but I'll lose most of the audience.

Pieter Abbeel

Well, I'm curious. I think it'd be great to hear it. But maybe we can zoom out for a moment before zooming back in. You talk about current methods use end to end learning backpropagation to power the end to end learning and you're saying switch to learn from less data and extract more from less data is going to be key as a way to make progress, to get closer to how the brain learns.

Geoff Hinton

Yes. So you get much bigger bandwidth for learning by having many, many little local objective functions.

Pieter Abbeel

And when we look at these local objective functions like filling in a blanked out part of an image or maybe filling back in a word, if we look at today's technologies, in fact, this is the current frontier and you've contributed. A lot of people are working exactly on that problem of learning from unlabeled data, effectively, because it costs a lot less human labor, but they still use backpropagation. The same mechanism.

Geoff Hinton

What I don't like about the masked auto-encoder is you have your input patches and then you go through many layers of representation. And the output of the net, you try to reconstruct the missing input patches. I think in the brain you have these levels of representation, but at each level you're trying to reconstruct what's at the level below. So it's not like you go through this many, many layers and then come back out again. It's that you have all these levels, each of which is trying to reconstruct what's at the level below. So I think that's much more brain-like. And the question is, can you do that without using backpropagation? Obviously, if you go through many, many levels and then reconstruct the missing patches with the output, you need to get information back through all those levels. And since we have backpropagation that's built into the simulators, you might as well do it that way. But I don't think that's how the brain is doing it.

Pieter Abbeel

And now imagine the brain is doing it with all these local objectives. Do you think for our engineered systems, will it matter whether, and sometimes there are three choices to make, it seems. One choice is what are the objectives? What are those local objectives that we want to optimize? A second choice is what's the algorithm to use to optimize it? And then a third choice is what's the architecture of how do we wire the neurons together that are doing this learning? And among those three, it seems like all three could be the missing piece that we're not getting right? Or what do you think?

Geoff Hinton

If you're interested in perceptual learning, I think it's fairly clear you want retinotopic maps, the hierarchy of retinotopic maps. So the architecture is local connectivity. And the point about that is, you can solve a lot of the credit design problem by just assuming that something in one locality and retinotopic map is going to be determined by the corresponding locality in the retinotopic map that feeds into it. So you're not trying to low down in the system, figure out how pixels determine what's going on a long distance away of the image. You're just going to use local interactions and that gives you a lot of locality and you'd be crazy not to use that. One thing neural nets do at present is they assume you're going to be using the same functions at every locality. So, convolutional nets do that. And transformers do that, too. I don't think the brain can do that because that would involve weight sharing. And it would involve doing exactly the same computation in each locality so you can use the same weights. I think it's most unlikely your brain does that. But actually, there's a way to achieve what weight sharing does, what convolutional nets do in the brain, in a much more plausible way than I think people have suggested before, which is if you do have contextual predictions, trying to agree with locally extracted things, then imagine a whole bunch of columns that are making local predictions, looking at nearby columns to get the contextual prediction. You can think of the context as a teacher for the local thing, but also vice versa. But think of the context of a teacher for what you're attracting locally. So you can think of the information that's in the context as being distilled into the local extractor. But that's true for all the local extractors. So what you've got is mutual distillation, where they're all trying teaching signals for each other. But what that means is knowledge about what you should extract in one

location is getting transferred into other locations, if they're trying to agree, if you're trying to get different locations to agree on something. If, for example, you find a nose and you find a mouth and you want them both to agree that they are part of the same face. So we should both give weights to the same representation. Then the fact that you're trying to get the same representation of different locations allows knowledge to be distilled from one location to another. And there's a big advantage of that over actual weight sharing. Obviously, biologically, one advantage is that the detailed architecture in these different locations doesn't need to be identical. But the other advantage is the front end processing doesn't need to be the same. So if you take your retina, different parts of the retina have different sized receptive fields and convolutional nets try to ignore that. They sometimes have multiple different resolutions and do convolution of each resolution, but they just can't deal with different frontend processing. Whereas if you're distilling knowledge from one location to another, what you're trying to do is get the same function from the optic array to the representation in these different locations. And it's fine if you pre-process the optic array differently in different locations. You can still distill the knowledge across the function from the optic array to the representation. Even though the frontend processing is different. And so although distillation is less efficient than actually sharing the weights, it's much more flexible and it's much more neurally plausible. So for me, that was a big insight I had about a year ago that we have to have something like weight sharing to be efficient. But local distillation will work if you're trying to get neighboring things to agree on a representation. But that idea of trying to get them to agree gives you the signal you need the knowledge in one location to supervise knowledge in another location.

Pieter Abbeel

And Geoff do you think what you're describing, one way to think of it is to say, hey, weight sharing is clever because it's something the brain kind of does, it just does it differently so we should continue to do weight sharing. Another way to think of it is that actually we shouldn't continue the weight sharing because the brain does it somewhat differently, and there might be a reason to do it differently. What's your thinking?

Geoff Hinton

I think the brain doesn't do weight sharing because it's hard for it to ship weights about the place. It's very easy if we're all sitting in a round. So I think we should continue to do convolutional things in convnets and in transformers. We should share weights. We should share knowledge by sharing weights. But just bear in mind that the brain is going to share knowledge not by sharing weights, but by sharing the function from input to output and using distillation to transfer the image.

Pieter Abbeel

Now there is the other topic that is talked about quite a bit, where the brain is drastically different from our current neural nets. And it's the fact that neurons work with spiking signals and that's very different from our artificial neurons in our GPUs. And so I'm very curious and your thinking on that, is that just an engineering difference or do you think there could be more to it that we need to understand better? And benefits to spiking?

Geoff Hinton

I think it's not just an engineering difference. I think once we understand why that hardware is so good. Why you can do so much in such an energy efficient way with that kind of hardware. We'll see that it's sensible for the brain to use spiking neurons. The retina, for example, doesn't use spiking neurons, the does lots of processing in the non spiking neurons. So once we understand why the cortex is using those, we will see

that it was the right thing for biology to do. And I think that's going to hinge on what the learning algorithm is, how you get gradients for networks of spiking neurons. At present, nobody really knows. At the present what people do is say, you see, the problem with a spiking neuron is there are two different, quite different kinds of decisions. One is exactly when does it spike? And the other is does it or doesn't it spike? So this discrete decision should be new on the spike or not, and then this continuous variable of exactly when it should spike. And people try to optimize the systems have come up with various kind of surrogate functions which sort of smooth things a bit so you can get continuous functions. They don't seem quite right. It'd be really nice to have a learning algorithm. When, in fact, in NuerIPs in about 2000, Andy Brown and I had a paper on trying to learn spiking Boltzmann machines. But it'd be really nice to get a learning algorithm that's good for spiking neurons. And I think that's the main thing that's holding up spiking neuron hardware. So people like Steve Furber in Manchester realized that, and many other people have realized that you can make more energy efficient hardware this way, but they built great big systems. Well what they don't have is good learning outcomes, which I think until we've got a good learning algorithm for it, we won't really be able to exploit what you can do with spiking neurons. And there's one obvious thing you can do with them that isn't easy in conventional neural nets. And that's agreement. So if you take a standard artificial neuron, when you simply ask the question, can you tell if its two inputs have the same value? But it can't. It's not an easy thing for a standard neuron to do, a standard artificial neuron. But if you're spiking neurons, it's very easy to build a system where if the two spikes arrive at the same time, they'll make their own fire. If they arrive at different times, they won't. So using the time of a spike seems like a very good way of measuring agreement. We know the biological system does that. You can see the direction the sound is coming from or hear the direction sound is coming from by the time delay in the signal between the two ears. And if you take a foot, that's about a nanosecond for light and it's about a millisecond for sound. And the point is, if I move something sideways in front of you by a few inches, the difference in the time to each of the two ears, the length of the path to the two ears, is only a small fraction of an inch. And so it's only a small fraction of a millisecond difference in the time the signal gets to the two ears. And we can deal with that and owls can deal with it even better. And so we're measuring, we're sensitive to times of 30 microseconds. In order to get stereo from sound. I can't remember what else is sensitive, but I think it's a lot better than 30 microseconds. And we do that by having two axons with spikes traveling in different directions, one from one ear, one from the other ear. So then you have cells fire, the spikes at the same time. That's a simplification but roughly that. So we know the spike timing can be used for exquisitely sensitive things like that. I would be very surprised if the precise spikes times wasn't being used. But we really don't know. And for a long time I thought it'd be really nice if you could use spiked times to detect agreement for things like self-supervised learning. Or for things like, if I've extracted your mouth, or extracted your nose or other representations of them. And from your mouth, I can predict something about your whole face. And from the nose, I can predict something about your whole face. But if your mouth and nose bring the right relationship to make the faces, predictions will agree. And it'd be really nice to have spike times to see if those predictions agree. But it's hard to make that work. And one of the reasons it's hard to make that work is because we don't know, we don't have a good algorithm for training in spiking neurons. So that's one of the things I'm focused on. How can we get a good training on networks of spiking neurons? And I think that'll have a big impact on hardware.

Pieter Abbeel

And that's a really interesting question you're putting forward there, because I doubt too many people are working on that compared to, let's say the number of people working on large language models or other problems that are much more, I guess, visible in terms of progress recently.

Geoff Hinton

Yeah, it's always a good idea to figure out what huge numbers of very smart people are working on and to work on something else.

Pieter Abbeel

Yeah, I think the challenge of course for most people, I'd say, including myself, but I definitely hear the question from many students too, is that it's easy to work on something else than everybody else, but it's hard to make sure that something else is actually relevant.

Geoff Hinton

Ah, yes.

Pieter Abbeel

Because there's many other things out there that are not very relevant you could possibly spend time.

Geoff Hinton

That involves having good intuitions.

Pieter Abbeel

Yeah. Listening to you, for example, could help. So I have a follow up question to something you just said, Geoff, which is that the retina doesn't use all spiking neurons. Are you saying that the brain has two types of neurons, some that are more like our artificial neurons and some that are spiking neurons.

Geoff Hinton

I'm not sure the retina is more like our artificial neurons, but certainly the cortex has, the neocortex, has spike neurons, and that's its primary mode of communication is by sending spikes from one parameter to another parameter in the cell. And I don't think we're going to understand the brain until we understand why it chooses to spend spikes. For a while I thought I had a good argument that didn't involve the precise time to spike. The argument went like this, the brains in the regime, where it's got lots and lots of parameters and not much data, relative to the typical neural nets we use. And there's a big danger of overfishing in that regime, unless you use very strong regularization. And a good regularization technique is a drop out, where each time you use a neural net, you ignore a whole bunch of the units. And so maybe the fact that the neurons are sending spikes, what they're really communicating is the underlying poisson rate. So let's assume this is poisson, which is close enough to this argument. There's a poisson process which sends spikes to category, but the rate of that process varies, and that's determined by the inputs of neurons. And you might think you'd like to send the real value of the rate from one unit to another. But if you want to do lots and lots of regularization, you can send the real valued rate with some noise added. And when we tried noise, just use spikes. That'll add lots of noise. And so this was the motivation for dropout. That the, most of the times, most of the neurons aren't involved in things, if you look at a fine time window. And you can think of spikes as a representation of ongoing poisson rage, it's just a very, very noisy representation, which sounds like a very, very bad idea because it's very, very noisy. But actually, once you understand about regularization of too many parameters, it's a very, very good idea. So I still have a lingering fondness for the idea that actually we're not using spike timing at all. It's just about using very noisy representations of

poisson rates to be a good regulator and I sort of flip between. I think it's very important when you do science, not to totally commit to one idea and ignore all the evidence for other ideas. But if you do that, you end up flipping between ideas every few years. So some years, I think neural nets are deterministic. I mean, we should have deterministic neural nets and that's what backprops using. Another year, I think it's about a five year cycle. I think no, no. It's very important that they're stochastic and that changes everything. So Boltzmann machines were intrinsically stochastic and that was very important to them. But the main thing is not to fully commit to either of those, but to be open to both.

Pieter Abbeel

Now, one thing, if we think more about what you just said, the importance of spiking neurons and figuring out how to train a spiking neural network effectively, what if we for now just say, let's not worry about the training part. Given that similarly it's far more power efficient, wouldn't people want to distribute pure inference chips that you pre-train effectively, separately and then you compile it onto a spiking neuron chip to have very low power inference capabilities. What about that?

Geoff Hinton

Yeah, so lots of people have thought of that and it's a very sensible idea and it's probably on the evolutionary path to getting to use spiking neural nets because once you're using them for inference and it works and there's people already doing that. It's already working and being shown to be more power efficient. And various companies have produced these big spiking systems. Once you're doing it for inference anyway, you'll get more and more interested in how you could learn in a way that makes more use of the available power in these spike times. So you can imagine a system where you learn using backdrop, but not on analog hardware, for example, this low energy hardware. And then you transfer to the lower energy hardware and that's fine. But we'd really like to learn directly in the hardware.

Pieter Abbeel

Now, one thing that really strikes me, Geoff, is when I think about your talks back around 2005, six, seven, eight when I was a PhD student, essentially pre AlexNet talks, those talks, I think topically, have a lot of resemblance to what you're excited about now. And it almost feels like AlexNet is an outlier in your path. How did you go from thinking so closely to how the brain might work to deciding that, you know, maybe you can first explain what was AlexNet? But also how did it come about? And what was that path to go from working on restricted Boltzmann machines, trying to see how the brain works to I would say that the more traditional approach to neural nets that you all of a sudden showed can actually work?

Geoff Hinton

Well, if you're an academic, you have to raise grant money, and it's convenient to have things that actually work, even if they don't work the way you're interested in them. So part of it is that, just go with the flow and if you can make backprop work well. And back then, in 2006, 2005, I got fascinated by the idea you could use stacks of restricted Boltzmann machines to pre-train feature detectors, and then it would be much easier to get backprop to work. It turned out with enough data, which is what you had in speech recognition and later on, because of Fei-Fei Li and her team in image recognition. With enough data, you don't need the pre-training, although pre-training is coming back. I mean, GPT-3 has pre-training. And pre-training is a thoroughly good idea. But once we discovered that we need to pretrain and that will make backprop work better and that did great things for speech, which George Dahl and Abdel-Rahman Mohamed did in 2009. Then Alex, who was a graduate student in my group then started applying the same ideas to vision. And

pretty soon we discovered that you didn't actually need this pre-training, especially if you have the ImageNet data. And in fact, that project was partly due to Ilya's persistence. So I remember Ilya coming into the lab one day and saying, look, now that we got speech recognition working. This stuff really works. We've got to do ImageNet before anybody else does. And retrospective around that, Yann LeCun was going into the lab and saying, look, we've got to do ImageNet with convnets before anybody else does. And Yann's students and some postdocs said, oh, I'm busy doing something else, so he couldn't actually get someone to commit to it. And Ilya initially couldn't get people to commit to it. And so Ilya persuaded Alex to commit to it by pre-processing the data for him. So he didn't have to pre-process the data. The data was all pre-processed to be just what he needed. And then Alex really went to town and Alex is just a superb programmer. And it was Alex who was able to make a couple of GPUs really sing. He'd make them work together in his bedroom at home. I don't think his parents realized that they were paying for most of the costs because that was the electricity. But he did a superb job approving convolutional nets on them. So Ilya said we've got to do this and helped Alex with the design and so on. Alex did the really intricate programing and I provided support and a few ideas about using dropout. I also did some good management. I'm not very good at management, but I am very proud of the management I did, which is with Alex Krizhevsky had to write a depth oral? to show that he was sort of capable of understanding research, which is what you have to do after a couple of years to stay in the PhD program. And he doesn't really like writing and he didn't really want to do the depth of it, but it was way past the deadline and the department was hassling us. So I said to him, each time you can improve the performance by 1% on ImageNet, you can delay your depth oral by another week. And Alex delayed his depth oral by a whole lot of weeks.

Pieter Abbeel

Yeah. And just for context, I mean, a lot of researchers know this, of course, but maybe not everybody. Alex's result with you and Ilya cut the error rate in half compared to prior work on the ImageNet and image recognition competition.

Geoff Hinton

More or less, I used to be a professor so it wasn't quite in half close. It cut it from about 26% to about 16 or 15%, depending on how you take it. It didn't cut it in half, but almost.

Pieter Abbeel

Almost in half. Whereas in previous years the progress was by 1% or 2%. Here was a very, very, whole different, well, that's why everybody switched from what they were doing, which was hand engineered approaches to computer vision, trying to program directly how can a computer understand what's in an image to deep learning.

Geoff Hinton

Now I should say one thing that's important to say here, Yann LeCun spent many years developing convolutional neural nets and it really should have been him, his lab to develop that system. We had a few little extra tricks, but they weren't the important thing. The important thing was to apply convolutional nets using GPUs to a big dataset. So Yann was kind of unlucky in that he didn't get the win on that, but it was using many of the techniques that he developed.

Pieter Abbeel

It didn't have the Russian immigrants that Toronto and you had been able to attract to make it happen.

Geoff Hinton

But one is Russian and one is Ukrainian and it's important not to confuse those. Even though the Ukrainian is a Russian speaking Ukrainian, don't confuse Russians with Ukrainians.

Pieter Abbeel

Absolutely.

Geoff Hinton

It's a different country.

Pieter Abbeel

So now, Geoff, that moment actually also marked a big change in your career because as far as I understand, you'd never been involved in corporate work. But it marked a transition for you, soon thereafter, from being a pure academic to being, ending up at Google, actually. Can you say a bit about that? How was that for you? Like, did you have any internal resistance?

Geoff Hinton

Well, I can say why that transition happened. What triggered it.

Pieter Abbeel

Yeah. I'm curious.

Geoff Hinton

So I have a disabled son who needs future provisions. So I needed to get a lump of money. And I thought one way I might get a lump of money was by teaching a Coursera course. And so I did a Coursera course on your neural networks in 2012. And it was one of the early Coursera courses, so the software wasn't very good. So it's extremely irritating to do. It really was very irritating and I'm not very good on software, so I didn't like that. And from my point of view, it amounted to, you agree to supply a chapter of a textbook, one chapter every week. So you had to give them these videos and then a whole bunch of people were going to watch the videos. Like sometimes the next day, Yoshua Bengio would say, why did you say that? So you know that it's going to be people who know very little, but also people know a whole lot. And so it's stressful. You know that if you are going to make mistakes, they're going to be caught, not like a normal lecture where you can just sort of press on the sustaining pedal and sort of fly your way through it, if you get so confused about something. Here, you have to get it straight. And the deal with the University of Toronto, originally, was that if any money was made from these courses, which I was hoping that there would be, the money that came to the university and would be split with the professor. They didn't specify exactly what the split would be, but one assumed it would be like 50-50 or something like that. And I was okay with that. The university didn't provide any support in preparing videos and then after I'd started the course and when I could no longer back out of it, the Provost made a unilateral decision without consulting me or anybody else, that actually, if money came from Coursera, the university would take all the money and the professor would get zero, which is exactly the opposite of what happens with textbooks. And the process was very like a textbook. I actually asked the university to help me prepare the videos and the AV people came back to me

and said, do you have any idea how expensive it is to make a video? And I actually did have an idea because I've been doing it. So I got really pissed off with my university because they unilaterally, sort of, canceled the idea to get any remuneration for this. They said it was part of my teaching, well actually, it wasn't part of my teaching. It was clearly based on lectures I had given as part of my teaching. But I was doing my teaching, as well as that, and I wasn't using that course for my teaching. And that got me pissed off enough that I was willing to consider alternatives to being a professor. And at that time, we then suddenly got interest from all sorts of companies in recruiting us, either in funding, giving big grants or in funding a startup. It was clear that a number of big companies were very interested in getting in on the act. And so normally I would have just said, no, I get paid by the state for doing research. I didn't want to try and make extra money from my research. I'd rather get on with the research. But because that particular experience with the university cheating me out of the money, now it turned out they didn't cheat me out of anything because no money came from the course anyway. But that pushed me over the edge into thinking, well okay, I'm going to find some other way to make some money. That was the end of my principles.

Pieter Abbeel

Oh, no. Well, but the result is that these companies are and in fact, if you read the Genius Maker's book by Cade Metz, which I reread last week in preparation for this conversation. If you read the book, it starts off with, actually, you running an auction for these companies to try to acquire your company, which is quite the start of a book. Very intriguing. But how was it for you?

Geoff Hinton

Oh, when it was happening, it was at NIPS. And Terry had organized NIPS in a casino at Lake Tahoe. And so in the basement of the hotel, there were these smoke-filled rooms full of people pulling One Armed Bandits and big lights flashing, saying you won $25,000 and all that stuff and people gambling in other ways. And upstairs we were running this auction. And we felt like we were in a movie. We felt like this was like being in that movie, The Social Network. It sort of felt like that. It was great. The reason we did it was that we had absolutely no idea how much we were worth. And I consulted a lawyer who said, there's two ways to go about this, you could hire a professional negotiator. In that case, you'll end up working for a company, but they'll be pissed off with you. Or you could just run an auction. As far as I know, this was the first time a small group like that just ran an auction. We run it on Gmail. I'd worked at Google over the summer, so I knew enough about Google to know that they wouldn't read our Gmail. And I'm still pretty confident they didn't read our Gmail. Microsoft wasn't so confident. And we just ran this auction where people had to Gmail me their bids and we then immediately mailed them out to everybody else with the timestamp of the Gmail. And then it just kept going up by half a million dollars, I think it was half a million to begin with then a million after that. And yeah, it was pretty exciting. Then we discovered we were worth a lot more than we thought. Retrospectively, we could probably have got more, but we got to a number that we thought was astronomical. And then basically we wanted to work for Google. So we stopped the auction so we could be sure of working for Google.

Pieter Abbeel

And as I understand it, you're still at Google today.

Geoff Hinton

I'm still at Google today, nine years later. I'm in my 10th year there. I think I'll get some kind of award when I've been there for ten years because it's so rare. Although people tend to stay at Google longer than other

companies. And yeah, I like it. But the main reason I like it is because the brain team is a very nice team and I get along very well with Jeff Dean. He's very smart, but very straightforward to deal with. And what he wants me to do is do what I want to do, which is basic research. He thinks what I should be doing is trying to come up with radical new algorithms. And that's what I want to do anyway. So it's just a very nice fit. I'm no good at managing a big team to improve speech recognition by 1%.

Pieter Abbeel

Well, it's better to just revolutionize the field again, right?

Geoff Hinton

Yeah. I would like to do it one more time. That's a bit ambitious, but.

Pieter Abbeel

I'm looking forward to it. I wouldn't be surprised at all. Now, when I look at your career, Geoff, and someone this information actually comes from the book as I didn't notice before when I had read the book the first time. I mean, you were a computer science professor at the University of Toronto. But computer science, you never got a computer science degree. You got a psychology degree. And you actually, at some point were a carpenter. How does that come about? How do you go from studying psychology to becoming a carpenter to getting into AI? What's the path for you there? How do you look at that?

Geoff Hinton

I'm in my last year at Cambridge. I had a very difficult time and got very unhappy and I dropped out. Just after the exams, I dropped out and became a carpenter. And I'd always enjoy carpentry more than anything else. So at high school there would be all the classes, and then you could stay in the evenings and do carpentry. And that's what I really looked forward to. So I became a carpenter. And then after being a carpenter for about six months, you couldn't actually make a living as a carpenter. So as a carpenter and decorator, I made the money doing decorating but I had the fun doing carpentry. The point is, carpentry is more work than it looks and decorating is less work than it looks. So you can charge more per hour for decorating. Unless you're a very good carpenter. And then I met a real carpenter. And I realized I was completely hopeless at carpentry. He was making a door for a basement, for a coal cellar under the sidewalk. And he was taking pieces of wood and arranging them so that they would warp in opposite directions so that it would cancel out. And that was a level of understanding and thought about the process that had never occurred to me. He could also take a piece of wood and just cut it exactly square with a hand saw. And he explained something useful to me. He said, if you want to cut a piece of wood square, you have to line the sole bench up with the room. Then you have to line the piece of wood up with the room. You can't cut it square if it's not aligned with the room. Which is very interesting in terms of coordinate frames. So anyway, because I was so hopeless compared to him, I decided I might as well go back into AI.

Pieter Abbeel

Now when you say get back into AI, as I understand it, this was at the University of Edinburgh where you went for a PhD?

Geoff Hinton

Yeah. I went to do a PhD there. But I went to neural networks with an eminent professor, Christopher Longuet-Higgins. He was really very brilliant. He almost got a Nobel Prize when he was in his thirties for figuring out something about borohydride. And I still don't understand what it is because it had to do with quantum mechanics, but it hinged on the fact that 360 rotation is not the identity operate, it's 720 degrees. He was interested in neural nets and the relation to holograms. And by the day I arrived, he lost interest in neural nets because he read Winograd's thesis and he became completely converted. He thought neural nets was the wrong way to think about it. We should do symbolic AI. He was very impressed by Winograd's thesis. He had a lot of integrity. So even though you completely disagree with what I was doing, he didn't stop me from doing it. He kept trying to get me to do stuff like Winograd's thesis. But he let me carry on doing what I was doing. And yeah, I was a bit of a loner. Everybody else back then, in the early seventies, was saying that's nonsense. Why are you doing this stuff? It's crazy. And in fact, the first talk I ever gave to that group was about how to do true recursion with neural networks. So this was a talk in 1973, so 49 years ago. And one of my first projects, I discovered a write up of it recently, was that you want a neural network that will be able to draw shape. And you want it to pass the shape into parts and you want it to be possible for a part of the shape to be drawn by the same neural hardware as the whole shape being driven by. So the neural hardware drawing the whole shape has to remember where it's got to in the whole shape and what the orientation and position and sizes for the whole show. But now it has to go off and you want to use the very same neurons for drawing a part of the shape. So you need somewhere to remember what the whole shape was and how far you've got in it so that you can pop back to that, once you've finished doing this subroutine, this part of the shape. So the question is, how is a neural network going to remember that? Because obviously you can't just copy the neurons. So I managed to get a system working where the neural network remembered it by having fast weights that we're just adapting all the time. And we're adapting so that any state that you've been in recently could be retrieved by giving it part of that state and then say, fill in the rest. And so I had a neural net that was doing true recursion, using the same neurons and the same weights to do the recursive call. And that was in 1973. And the, I think people didn't understand the talk because I wasn't very good at giving talks. But they said why would you want to do recursion to the neural nets? You can do recursion with LISP. They didn't understand the point, which is that unless you can get neural nets to do something like recursion, we're never going to be able to explain a whole bunch of things. And now that's become sort of an interesting question, again. So I'm going to wait one more year until that idea is an antique. A genuine antique, it would be 50 years old. And then I'm going to sort of write up the research I did. And it was all about fast weights as a memory.

Pieter Abbeel

So I have many questions here, Geoff. The first one is, you're standing in this room where everybody is, you're a PhD student, or maybe fresh out of PhD, you're standing in a room with essentially everybody telling you what you're working on is a waste of time. And you were convinced somehow it was not. Where do you get that conviction from?

Geoff Hinton

I think a large part of it was my school. So my father was a communist. But he sent me to an expensive private school because they had a good science education. And I was there from the age of seven, they had a preschool. And it was a Christian school. And all the other kids believed in God. And it was just, at home, I was told that that was nonsense. And it did seem to me that it was nonsense. And so I was used to just having everybody else being wrong and obviously wrong. And I think that's important. I think you need, I respect that you need faith, which is funny in this situation, you need the faith in science to be willing to work

on stuff just because it's obviously right. Even though everybody else says it's nonsense. And in fact, it wasn't everybody else. It was everybody else in the early seventies doing AI saying it was nonsense, for nearly everybody else. But if you look a bit earlier, if you look in the fifties, both Von Neumann and Turing believed in neural nets. Turing, in particular, believed in neural nets training for reinforcement learning. If I still believe if they hadn't both died early, the whole history of AI might have been very different because they were sort of powerful enough intellects to have swayed the field. And they were very interested in how the brain works? So I think it's just bad luck they both died early. Well, British intelligence might've come into it, but.

Pieter Abbeel

Now you go from believing in this when, at the time, many people didn't to getting the big breakthroughs that helped power almost everything that's being done today. And now there is in some sense, the next question is, it's not just that deep learning works and works great. The question becomes, is it all we need or will we need other things? And you've said things, maybe I'm not literally quoting you, but to the extent of deep learning will do everything.

Geoff Hinton

What I really meant by that, I sometimes say things without thinking, without being accurate enough. And when people call me on it, like saying we won't need radiologists. So what I really meant was using stochastic gradient descent with just a whole bunch of parameters. That's what I sort of had in mind when I said deep learning. The way you get the gradient might not be backpropagation and the thing you get the gradient of might not be some final performance measure, but rather these lots of local objective functions. But I think that's how the brain works, and I think that's going to explain everything else, yes.

Pieter Abbeel

Well, nice to see it confirmed.

Geoff Hinton

So one other thing I want to say is the kind of computers we have now are very good for doing banking because they can remember how much you have in your account. It wouldn't be so good if you went in with, well, you got roughly this much. We're not really sure because we don't do it to that precision, but roughly this much. So we don't want that in a computer doing banking or on a computer guiding the space shuttle or something. We'd really rather it got the answer exactly right. And they're very different from us. And I think people aren't sufficiently aware that we made a decision about how computing would be, which is that a computer, our knowledge would be immortal. So if you look at existing computers, you have a computer program or maybe you just have a lot of weights for neural nets. That's a different kind of program. But if your hardware dies, you can run the same program on another piece of hardware. And so that makes the knowledge immortal. It doesn't hinge on that particular piece of hardware surviving. Now the cost of the immortality is huge because it means the two different bits of hardware have to do exactly the same thing. Obviously there is corruption, all that, but if you don't own the error, you have to do exactly the same thing, which means they better be digital, mostly digital. And they're probably going to do things like multiple numbers together, which involves using lots and lots of energy to make things very discrete, which is not what hardware really wants to be. And so as soon as you commit yourself to the immortality of your program or your neural net, you're committed to very expensive computations and also to very expensive manufacturing processes. You need to manufacture these things accurately, probably in 2D and then put lots

of 2D things together. But if you're just willing to give up on immortality. Sort of in fiction, normally what you get in return is love. But if we're willing to give up immortality, what we'll get in return is very low energy computation and very cheap manufacturing. So instead of manufacturing computers, what we should do is grow them. We should use nanotechnology to just grow the things in 3D. And each one will be slightly different. So the image I have is if you take a pot plant and you sort of pull it out of its pot, there's a root bowl, but it's the shape of the pot. And so all the different pot plants have the same shape root bowl, but the details of the roots are all different, but all doing the same thing, extracting nutrients from the soil. They got the same function and they're pretty much the same. But the details are all very different. So that's what real brains are like. And I think that's what I call mortal computers will be like. So these are computers that are grown rather than manufactured. You can't program. They just learn. They obviously have a learning algorithm sort of built into them. They learn. They can do most of their computation in analog. Because analog is very good for doing things like taking a voltage times the resistance and turning it into a charge and adding the charge. And there are already chips that do things like that. The problem is, what do you do next? And how do you learn in those chips? And at present, people have suggested backpropagation or various versions of Boltzmann machines. I think we're going to need something else. But I think sometime in the not too distant future, we're going to see mortal computers, which are very cheap to create, have to get all the knowledge in there by learning at a very low energy. And these mortal computers, when they die, they die and their knowledge dies with them. And so, it's no use looking at the weights because those weights only work for that hardware. So what you are going to do is distill the knowledge into other computers. So when these mortal computers get old, they're going to have to do lots of podcasts to try and get the knowledge into younger model computers.

Pieter Abbeel

And the first one you build, I'll happily have that one on. Let me know. So Geoff, this reminds me of another question that has been on my mind for you, which is when you think about today's neural nets, the ones that grab the headlines are very, very large. I mean, not as large as the brain maybe, but in some sense starting to get to that size. The large language models. But the results look very, very impressive. So, one, I'm curious about your take on those kinds of models and what you see in them? And what you see in their limitations? But two, I'm also curious about what do you think about working on the other end of the spectrum? For example, ants have much smaller brains, obviously, than humans, yet it's fair to say that our visual motor systems that we have developed artificially are not yet at the level of what ants can pull off or bees and so forth. And so I'm curious about that spectrum as well as the recent big advances in language models, what do you think about those?

Geoff Hinton

So bees, they may look small to you, but I think a bee has about a million neurons. So I think a bee is closer to GPT-3, certainly closer than an ant is. But a bee is actually quite a big neural net. My belief is that if you take a system with lots of parameters and they choose sensibly using some kind of gradient descent in some kind of sensible, objective function, then you'll get wonderful properties out of it. You'll get all these emerging properties, like you do with GPT-3 and also be the Google equivalents that I've talked about so much. That doesn't sort of settle the issue of whether they're doing it the same way as us. And I think we're doing a lot more things like recursion, which I think we do in neural nets. And I try to address some of these issues in the paper I put on the web last year called GLOM, well I call it GLOM. It's how you do part-whole hierarchies in neural nets. So you definitely have to have structure. And if what you mean by symbolic computation is just that you have a part-whole structure, then we do symbolic computation. That's not

normally what people mean by symbolic computation. The sort of hard line symbolic computation means you're using symbols and you're operating all symbols using rules that just depend on the form of the symbol during your processing. And that a symbol, the only property a symbol has is that is either identical or not identical to some other symbol, and perhaps that it points to something, can be used as a pointer to get to something. The neural nets are very different from that. So the sort of hard line simple processing, I don't think we do that, but we certainly deal with some part-whole hierarchies. But I think we do it in great big neural nets. And I'm sort of up in the air at present as to what extent does GPT-3 really understand what it's saying? I think it's fairly clear, it's not just like the old ELIZA program, which just rearranges trainings of symbols that had no clue what it was talking about. And the reason to believe that, as you say in English, show me a picture of a hamster wearing a red hat. And it draws a picture of a hamster wearing a red hat. And you're fairly sure it never got that pair before. So it has to understand the relationship between the English training and the picture and before it've done that, if you've asked any of these doubters, these neural net skeptics or neural net deniers. Let’s call them neural net deniers. If you'd asked them, well, how would you show that it understands? I think that I've accepted that, well, if you want to draw pictures on something and it draws a picture of that thing, then it understood. Just as was Winograd's thesis, you ask it to put the blue block in the green box and it puts the blue block in the green box. And so that's pretty good evidence it understood what you said. But now that it does it, of course, the skeptics then say, well, you know, that doesn't really count. There's nothing that would satisfy them basically.

Pieter Abbeel

Yeah, the goal line is always moving for true skeptics. Now, there is the recent one, the Google one, the PaLM model that in the paper showed how it was explaining, effectively how jokes work.

Geoff Hinton

That was extraordinary, right?

Pieter Abbeel

That just seemed very deep understanding of language.

Geoff Hinton

No, it was just rearranging the words you had trainings of.

Pieter Abbeel

You think so?

Geoff Hinton

No. No. I don’t see how it could generate those explanations without sort of understanding what's going on. Now, I'm still open to the idea that because it was training backpropagation it could’ve ended up with a very different sort of understanding from us. And obviously adversarial images tell you a lot that you can recognize objects by using their textures and you can be correct about it in the sense of, or generalize to other instances of those objects. But it's a completely different way of doing it from what we do. And I like to think of the example of insects and flowers. So insects see in the ultraviolet. So two flower that look the same to us can look completely different to insects. And now because the flowers look the same to us, do we say the insects are getting it wrong? Because these flowers evolved with the insects to give signals to

the insects in the ultraviolet to tell them which flower it is. So it's clear the insects are getting it right and we just can't see the difference. And that's another way of thinking about adversarial examples. It looks, you know, this thing that it says is an ostrich looks like it looks like a school bus to us. But actually, if you look in the texture domain, then it's actually an ostrich. So the question is who is right? In the case of the insects, just because two flowers ook identical to us, it doesn't mean they're really the same. The insects are right about them being very different. In that case it's different parts of the electromagnetic spectrum, indicating differences that we don't pick up on. But it could be textures.

Pieter Abbeel

In the case of image recognition for our current neural nets though, you could argue maybe that since we build them and we want them to do things for us in our world that we really don't want to just say, okay, they got it right and we got it wrong. I mean, they need to recognize the car and the pedestrian.

Geoff Hinton

Yeah, I agree. I just want to show it's not as simple as you might think about who's right and who's wrong. And part of the point of my GLOM paper was to try and build perceptual systems that work more like us. So they're much more likely to make the same kinds of mistakes as us and not make very different kinds of mistakes. But obviously, if you've got a self-driving car, for example, if it makes a mistake that any normal human driver would have made, that seems much more acceptable than making a really dumb mistake.

Pieter Abbeel

So, Geoff, as I understand it, sleep is something you also think about. Can you say a bit more?

Geoff Hinton

Yes. I often think about it when I'm not sleeping at night. There's something funny about sleep, which is some animals do it, fruit flies sleep. And it may just be to stop them flying around in the dark. But if you deprive people of sleep. Then they go really weird, like if you deprive someone for three days, they start hallucinating. If you deprive someone for a week, they go psychotic and may never recover. These are nice experiments done by the CIA. And the question is, one, why do we, what is the computational function of sleep? There is presumably a pretty important function for it if depriving you, it makes you just completely fall apart. So current theories are things like it's for consolidating memories or maybe for downloading things from the hippocampus into cortex, which is a bit odd since age come from cortex and hippocampus in the first place. So a long time ago, in the early eighties, Terry Sinofsky and I had this theory called Boltzmann machines. And it was partly based on an insight of Francis Crick when he was thinking about Hopfield nets, Francis Crick and Graeme Mitchinson had a paper about sleep and the idea that you would hit the net with random things and tell it not to be happy with random things. So with Hopfield nets you give it something you wanted to memorize and it changes the weights so the energy of that vector is low. Well, the idea is if you also give it random vectors and say make the energy high, the whole thing works better. And that led to Boltzmann machines, where we figured out that if you instead of giving it random things, you get things generated from a Markov chain, the model's own Markov chain, and you say make those less likely, make the data more likely, that is actually maximum likelihood learning. And so we got very excited about that because we thought, okay, that's what sleep is for. Sleep is this negative phase of learning. It comes up again, now, in contrastive learning, where you have two patches from the same image, you try to get them to have similar representations. And two patches from different images, you try to get them to have representations that are sufficiently different. Once they're different, you don't make them any more different,

but you stop them being too similar. And that's how contrastive learning works. Now, with Boltzmann machines, you couldn't actually separate the positive face from the negative face you had to interleave positive examples and negative examples. Otherwise, the whole thing would go wrong. And I tried a lot not interleaving them and it's quite hard to do a lot of positive examples followed by a lot of negative examples. What I discovered a couple of years ago that got me very excited, and caused me to agree to give lots of talks that I then canceled when I couldn't make it work better, was that with contrastive learning you can actually separate the positive and negative phases. So you can do lots of examples of positive pairs, followed by lots of examples of negative pairs. And that's great because what that means is you can have something like a video pipeline where you're just trying to make things similar while you're awake. And trying to make things dissimilar while you're asleep. If you can figure out how sleep can generate video for you. So it makes contrastive learning much more plausible if you can separate the positive and negative phases and do them at different times and do a whole bunch of positive updates for a whole bunch of negative updates. Even for the standard contrastive learning, you can do that moderately well. You have to use lots of momentum and stuff like that. There's also the little trick to make it work, but you can't make it work. So I now think it's quite likely that the function of sleep is to do unlearning on negative examples. And that's why you don't remember your dreams. You don't want to remember them. You're unlearning them. Crick pointed this out. You'll remember the ones that are in the fast weights when you wake up because the fast weights are a temporary store. So that's not unlearning. That still works the same way. But the long term memory, the whole point is to get rid of those things and that's why you dream for many hours a night. But when you wake up, you can just remember the last minutes of the dream you have when you wake up. And I think this is a much more plausible theory of sleep than any other I've seen, because it explains why if you got rid of it, the whole system would just fall apart. You could go disastrously wrong, and start hallucinating, and do all sorts of things. But let me say a little bit more about the need for negative examples when you have a contrastive learning. If you've got a neural net and it's trying to optimize some internal objective function, something about the kinds of representations that it has, or something about the agreement between contextual predictions and local predictions. It wants this agreement to be a property of the real data. And the problem inside a neural net is that you might get all sorts of correlations in your inputs. I'm a neuron, right? So I get all sorts of correlations in my inputs, and those correlations have nothing to do with the real data. They are caused by the wiring of the network and the weights in the network. If these two neurons are both looking at the same pixel, they'll have a correlation. But that doesn't tell you anything about the data. And so the question is, how do you learn to extract structure that's about the real data and not about the wiring of your network? And the way to do that is to feed it positive examples and say find structure in the positive examples that isn't in the negative examples because the negative is going to go through exactly the same wiring. And if the structure is not in the negative examples, but it is in the positive examples, then the structures are about the difference between the positive and negative examples, not about your wiring. So people don't think about this much. But if you have powerful learning algorithms, you better not make them learn about the neural networks' own weights and wiring. That's not what's interesting.

Pieter Abbeel

Now, when you think about people who don't get sleep then and start hallucinating, is hallucinating effectively trying to do the same thing, you're just doing it while you're awake?

Geoff Hinton

Obviously you can have little naps and that's very helpful. And maybe hallucinating when you're awake is serving the same function of sleep. And it's I mean, all the experiments I can say it's better to not have 16

hours awake and 8 hours asleep. It's better to have a few hours awake and a few hours asleep. So a lot of people have discovered that little naps help. Einstein used to take little naps all the time. And he did okay.

Pieter Abbeel

Yeah, he did very well, for sure. Now there's this other thing you've brought up, this notion of student beats teacher. What does that refer to?

Geoff Hinton

Okay, so a long time ago, I didn't experiment on MNIST, which is a standard digit database recognizing digits, where you take the data, the training data and you corrupt it. And you corrupt it by substituting the wrong label, one of the other nine labels, 80% of the time. So now you've got a dataset in which the labels are correct 20% of the time and wrong percent of the time. Then the question is, can you learn from that? And how well do you learn from that? Well, the answer is you can learn to get like 95% correct on it. So now you've got a teacher who's wrong 80% of the time, then the student is right 95% of the time. So the student is much, much better than the teacher. And this isn't each time you get an example you corrupt it, you take the training samples, you corrupt them once and for all. So you can't average away the corruption over different, you won't be able to get the average with different training cases and have similar images. But if you ask, well, how many training cases do you need if you have corrupted ones? And this was of great interest because of the tiny images taken some time ago where they had 80 million tiny images with a lot of wrong labels. And the question is, would you rather have a million things that are flakily labeled? Or would you rather have 10,000 things with accurate labels? And I had a hypothesis that what counts is the amount of mutual information between the label and the truth. So if the labels are correct 90% of the time there's no mutual information between the labels and the truth. If they are corrupted 80% of the time there's only a small amount of mutual information. I think it's, my memory is, is 0.06 bits per case, whereas if it's uncorrupted, it's about 3.3 bits per case. So it's only a tiny amount. And then the question is, well, suppose I balance the size of the training set by putting as much mutual information in there. So if there's like a 50th of the mutual information, I have 50 times as many examples, do I now get the same performance? And the answer is yes you do, within a factor of two. I mean, the training set actually needs to be twice that big, but roughly speaking, you can see how useful a training example is by the amount of mutual information between the label and the truth. But I noticed recently you have something for doing sim to real where you're labeling real data using a neural net and these labels aren't perfect. And then you take the student that learned from these labels and the student is better than the teacher it learned from. People are always puzzled by how the student could be better than the teacher. But in neural nets it is very easy. The student will be better than the teacher if there's enough training data, even if the teacher is very flaky. And I have a paper a few years ago with Melody Guan about this, for some medical data, but the first part, the paper talks about this. But the rule of thumb is basically what counts is the mutual information between the assigned label and the truth. And that tells you how valuable the training example is. And so you can make do with lots of flakey ones.

Pieter Abbeel

That's so interesting. Now in the work we did that you just referenced, Geoff, and in the work I've seen quite popular recently, usually the teacher provides noisy labels. But then not all the noisey labels are used. There is a notion to only look at the ones where the teacher is more confident. Your description doesn’t really care about that.

Geoff Hinton

That's obviously a good hack, but yeah, you don't need to do that. You don't need to do that. It's a good hack and it probably helps to only look at the ones where you have reason to believe the teacher got it right. But it'll work even if you just look at them all. And there is a phase transition. So with endless, Melody plotted graphs and as soon as you get like 20% of the labels right, your student will get like 95% correct. But as you get down to about 15% right, you suddenly get a phase transition where you don't do any better than chance because somehow the student has to get it. The teacher is saying these labels and the student has to, in some sense, understand which case is right and which cases are wrong. So see the relationship between the labels and the inputs. And then once the students in that relationship are wrongly labeled things, it's just very obviously wrong. So it's fine if it's randomly wrong, like. But there is a phase transition where you have to have a good enough so the students sort of get the idea. But that explains how our students are smarter than us.

Pieter Abbeel

We only need to get it right a small fraction of the time.

Geoff Hinton

Right. And I'm sure the students do some of this data curation where you say something, the student thinks, oh that's rubbish, I'm not going to listen to that. These are the very best students, you know.

Pieter Abbeel

Yeah, those are the ones that can surprise us. Now, one of the things that is really important in neural net learning, and especially when you're building models, is to get an understanding of what is it learning? And often people try to somehow visualize what's happening during learning. And one of the most prevalent visualization techniques is called t-SNE, which is something you invented Geoff. So I'm curious how did you come up with that? And maybe first describe what it does and then what's the story behind it?

Geoff Hinton

So if you have some high dimensional data and you try and draw a 2D or 3D map of it, you could take the first two principal components and just plot the first two principal components. But what principal components care about is getting the big distances right. So if two things are very different. Principal components are very concerned to get them very different, in the 2D space. It doesn't care at all about the small differences because this is sort of operating on the squares of the big differences. So it won't preserve similarity very well, high dimensional similarity. And you are often interested in just the opposite. You've got some data and you're interested in what's similar to what? You don't care if it gets a big distances a bit wrong as long as it gets the small distances right. So I had the idea a long time ago that what if we took the distances and we turned them into probabilities of pairs? There's various versions of t-SNE, but suppose we turned them into the probability of a pair such that we say pairs with a small distance are probable and the pairs with a big distance are improbable. So we converting distances into probabilities in such a way that small distances correspond to big probabilities. And we do that by putting a guess in around a point, your data points and computing the density of the other data point under this. And that's an on normalized probability that you normalize these things. And then you try and lay the points out in 2D as to preserve those probabilities. And so it won't care much if two points are far apart. They'll have a very low pairwise probability and it doesn't care the relative positions of those two points. What it cares about is the relative

positions of the ones with high probability. And that produces quite nice maps. And that was called Stochastic Neighbor Embedding. Because we thought of this, if we pick stochastically pick a neighbor according to the density of the Gaussian. And I did that work with Sam Roweis and it had very nice simple derivatives, which convinced me that we were onto something. And we got nice maps but they tended to crowd things together. And there's obviously a basic problem in converting high dimensional data into low dimensional data. So in, SNE tends to crowd things together in a stochastic neighbor embedding and that's because of the nature of high dimensional spaces and low dimension spaces. In a high dimensional space, a data point can be close to lots of other points without them all being too close to each other. In a low dimensional space, they will all have to be close to each other if they are close to this data point. So you've got a problem in embedding closeness from high dimensions to low dimensions. And I had the idea when I was doing SNE that since I was using probabilities as this kind of intermediate currency, there should be a mixture model. There should be a mixture version where you're saying high dimensions, the probability of a pair is proportional to e to the minus, squared distance a Gaussian. And in low dimensions, suppose you have two different maps. The probability of a pair is the sum of e to the minus of distance in the first 2D map and e to the minus of the squared distance on the second 2D map. And that way, if we have a word like bank, we're trying to put similar words near one another. Bank can be close to greed in one map and can be close to river in the other map, without river ever being close to greed. So I really pushed that idea because I thought it's a really neat idea and you can have a mixture of maps. And we managed to get, Ilya was one of our first people to work on it and James Cook worked on it a lot and several other students worked on it and we never really got it to work well. And I was very disappointed that somehow we aren't being able to make use of the mixture idea. And then I went to a simpler version, which I called UNI-SNE, which was a mixture of the Gaussian and a uniform. And that worked much better. So the idea is in one map, all pairs are equally probable. And that gives you a sort of background probability, which comes with the big distances, the small background probability. And then in the other map, you contribute a probability proportional to the square distance in this other map. But it means in this other map, things can be very far apart if they want to be. Because the fact that then they need some probability is taken care of by the uniform map. And then I got a review paper from Laurens van der Maaten, which I thought was actually a published paper because of the form it arrived in but it wasn't actually a published paper. And he wanted to come do research with me. I thought he had this published paper, so invited him to come do research. Turns out he was extremely good and it's lucky I'd be mistaken in thinking it was a published paper. And we started on UNI-SNE. And then I realized that actually UNI-SNE is a special case of using a mixture of a Gaussian and a very, very broad Gaussian, which is a uniform. So what if we use a whole hierarchy of Gaussians, many, many Gaussians with different weights, and that is called a t-distribution. And that led to t-SNE and t-SNE works much better. t-SNE has a very nice property that it can show you use things at multiple scales because it's got a kind of a 1 over d squared property that once distances get big it behaves just like gravity and clusters of galaxies and there are clusters of galaxies and galaxies and clusters of stars and so on. And you get structure at many different levels and you get the core structure and the fine structure all showing up. Now the objective function used for all this, which was the sort of relative density under a Gaussian, came from other work I did with Alberto Paccanaro, earlier, that we found hard to get published. I got a review saying, yeah, I got a review of that work when it was rejected by some conference saying Hinton's been working on this idea for seven years and nobody's interested. I take those reviews as telling me I'm on to something very original. And that actually had the function that's now used, I think its called NCE, it's used in these contrasting methods. And t-SNE is actually a version of function. But it's being used for making maps. So it's a very long history of t-SNE, of getting the original SNE, and then trying to make a mixture version. And it's just not working, not working, not working. And then eventually getting the coincidence of figuring out it was a t-

distribution is what we wanted to use, that was the kind of mixture. And Laurens arriving, and Laurens was very smart and very good programmer and made it work beautifully.

Pieter Abbeel

This is really interesting because it seems a lot of the progress these days, the bigger idea plays a big role. But here it seems it was really getting the details right was the only way to get it to fully work.

Geoff Hinton

You typically need both. You have to have a big idea for it to be interesting original stuff, but you also have to get the details right. And that's what graduate students are for.

Pieter Abbeel

So, Geoff. Thank you. Thank you for such a wonderful conversation.

www.therobotbrains.ai /geoff-hinton-transcript-part-two

Geoff Hinton Transcript Part Two | The Robot Brains

38-49 minutes

Geoff Hinton on The Robot Brains Season 2 Episode 22

Pieter Abbeel

In last week's episode, we had Geoff Hinton on the show. We covered so much ground from the early days when very few people were working on neural nets and deep learning, through the ImageNet, AlexNet breakthrough moment through Geoff's current work and vision for the future of AI. As you might recall, we also gave you an opportunity to contribute questions through Twitter. In today's episode, we'll discuss some of these questions with Geoff. But before we dive into our very last episode of season two, I just want to say it has been such a pleasure and honor to have so many amazing guests on the show this season. We had guests explaining how AI is being used in real businesses today, like for Flora Tasse on building AI for customer service, Amit Prakash on helping companies use AI to make better decisions, and Benedict Evans on what really matters about tech today. There were guests using AI to solve major health issues like George Netscher on using AI to protect the elderly with fall detection, Athelas' Tanay Tandon on using AI to improve blood testing, Andrew Song on how AI is helping give back hearing to people worldwide, and Param Hedge on using AI to improve training and prevent injuries in sport. We had guests using AI for social good like Ayanna Howard tackling bias in AI, Revolution Robotics' Jared Schreiber on teaching children about AI robotics and David Rolnick on using AI to fight climate change. We also had guests who are using AI in industry giants like Microsoft's Eric Horvitz on using air for the greater good and Shakir Mohamed from DeepMind on weather prediction. There were guests using AI and consumer applications like Spotify's Gustav Söderström on AI in delivering personalized experiences, Amit Aggarwal from THE YES on using AI to serve up a better experience in fashion and Etsy's Mike Fisher on AI in e-commerce. We had guests using AI and transportation and futuristic vehicles like Adam Bry on using AI to power Skydio drones and MIT's Cathy Wu on the future of our roads and Alex Kendall on Wayve's driverless cars. We had guests making AI accessible for all through open source like Ross Wightman and Hugging Face's Clement Delangue. And we kicked off and ended our series with academic leaders in the field like Sergey Levine from UC Berkeley on current research challenges in AI. And of course, last but by no means least, Geoff Hinton. Speaking of which, let's get to your questions for Geoff. Geoff, thank you for making the extra time for audience questions. It's actually the first time we've done this on the podcast and we had so many questions on Twitter for you. It's clear so many people want to learn from you and have questions for you. Hopefully we can get through a bunch of these questions. Let me kick it off with a question from somebody you are very familiar with, Ilya Sutskever. What were some of the hardest times, research wise, on the path to making deep learning work? Was there ever a time where it just wasn't clear how to even make the next step?

Geoff Hinton

I think it was always the case that there were things worth trying, even if they didn't work and consistently didn't work. I never reached a point where I thought there's just, I can't see where to go from here. There were always many possibilities, many leads to kind of follow up, most of which ended in dead ends. But I think good researchers always have, dozens of things they'd like to try. They just don't have time to try them.

So for me, there was never a point where I thought, it's completely hopeless. There's particular algorithms that at times I thought that's completely hopeless, like Boltzmann machine learning. Sometimes I think it's hopeless, sometimes I don't. But the whole enterprise of which, I could now phrase as, can you find objective functions and get the gradients so that you can learn by stochastic gradient descent? That whole enterprise always seemed to me to be, there were always directions you could go to push it forwards.

Pieter Abbeel

And I have a second question from Ilya, a very different question. Are you ever concerned that AI's becoming too successful and too dominant?

Geoff Hinton

Yeah. The two things that concern me most are it's use in weapons because that will allow countries like the United States, for example, to have little foreign wars with no casualties by using robot soldiers. I don't like that idea. Even worse, its use in targeting particular subpopulations to swing elections. So this kind of stuff was done by Cambridge Analytics and that I believe was very influential in both Brexit and the election of Trump. I think it's very unfortunate that techniques like deep learning are going to make that kind of operation more efficient.

Pieter Abbeel

Question from Pouria Mistani, is deep learning hitting a wall? Will AGI be achieved by scaling up neural activities in deep learning architectures?

Geoff Hinton

It won't be achieved just by scaling up neural numbers of parameters or neural connectivities, but it's not hitting a wall. I recognize where that quote comes from. It's a sort of attention grabbing quote. And this is regularly said that deep learning is hitting a wall and it regularly keeps making more progress. And if any of the people who say it's hitting a wall, would just write down a list of the things it's not going to be able to do. And then five years later, we'll be able to show we've done it.

Pieter Abbeel

I like that notion that anybody who wants to claim something is hitting a wall to make a list of things it cannot do. And that's great inspiration for all the rest of us to see if we can make it happen or not.

Geoff Hinton

But it has to be fairly well defined what it can do. Like there was Hector Levesque, who's a symbolic AI guy, and a very good one. Actually made a criterium, which is the Winograd sentences where you say things like the trophy would not fit in the suitcase because it was too small versus the trophy would not fit in the suitcase because it was too big. And if you want to translate that into French, you have to understand that in the first case it refers to suitcase, and in the second case it refers to trophy because there are different genders in French. And the only machine translation with neural nets was random. It couldn't get the gender right when it was in French. And it's getting better all the time. But at least Hector made a very clear definition of what it would mean for a neural net to understand what was going on. And we're not there yet, but I think we're considerably better than random. But I'd like to see more of that by people who are skeptics.

Pieter Abbeel

Great challenges, yeah. Next question is from Eric Jang, actually, one of your colleagues at Google. What are three questions that keep you up at night? Not necessarily restricted to machine intelligence.

Geoff Hinton

When is the attorney general finally going to do something? That keeps me up at night. Because time's running out. That's what I worry about most. How does the world deal with people like Putin who have nuclear weapons? And does the brain use backpropagation or not? In that order.

Pieter Abbeel

Love the contrast of that. Third one with the other two. Eric had another question, I'm going to go here. You spent years working on topics that the mainstream machine learning community thought was niche. What advice do you have for contrarians trying to produce the next AlexNet result?

Geoff Hinton

Just trust your own intuitions. I have the standard thing I say, which is maybe you got good intuitions or you haven't. If you haven't got good intuitions, it doesn't matter what you do. If you have good intuition, you trust him. But of course, that needs to be patted out with where do intuitions come from? And good intuitions come from a lot of hard work trying to understand things. And basically I think we're based on generality machines. So lots of experience with similar things is where intuitions come from. So you just need a lot of experience and then trust your intuitions.

Pieter Abbeel

Your next one comes from Danielle Newnham. What is the connection, as you see it, between mania and genius?

Geoff Hinton

Oh, that's very interesting. I'm slightly manic depressive, so I tend to oscillate between having very creative periods when I'm not very self-critical. And having mildly depressed periods when I'm extremely self-critical. And I think that's more efficient than just being kind of uniform. So what happens in manic periods is you just ignore all the problems. You're so sure there's something exciting here. Yeah, sure, there's all those obvious problems but don't let it stand in our way. Let's get on with it. But then when you're depressed, all these obvious problems overwhelm you. And the question is, can you keep going and sort them out and figure out whether the idea really was good or not? And I tend to sort of alternate like that, which is why every so often I tell people I figured out how the brain works, and then I go through a long period of figuring out why that isn't actually true, which is slightly depressing. I think it's just got to be like that. There's a poem by William Blake. It has a pair of lines they would go, joy and woe are woven fine, a clothing for the soul divine. And it's basically saying that's just the nature of being joy and woe together. And I think that's the nature of research, too. And if you don't get really excited and you don't get really fed up when it doesn't work, then you're not a real researcher. Well maybe you're just a different kind of researcher.

Pieter Abbeel

There's a related question as part of that, what childhood experiences shaped you the most and how?

Geoff Hinton

I think the most formative experience was coming from the home in which everybody was clear that religion was nonsense. And being sent to a private school, which was a Christian school where when I was seven, when I first went there, everybody believed in God. Everybody except me that was. That was a very formative experience for me, possibly because I got a large ego, I realized that everybody was wrong. But having that experience of seeing everybody else being wrong and gradually over the years, seeing them change their minds and seeing these teenage boys say, well, maybe, maybe God isn't real. That was very helpful.

Pieter Abbeel

Next one is from Bishal Binayak. What's your thought process to solve a research problem? Is it mainly focusing on machine learning? Probably implying maybe, you know, the question is, do you also need to think about other fields?

Geoff Hinton

Because we don't necessarily have good insights into our own thought processes. I guess I tend to work a lot with analogies, so at least I'm consistent, that is, I think the basic form of human reasoning is analogies which are based on having the right features and big vectors. And that's how I do research too. I try and look for similar things and maybe it's not so much try as similar things sort of pop into my mind and I think everything I'm doing is a kind of result of these analogies with many, many other things via these feature vectors where I'm sort of basically unaware of many of these analogies, but in a different way. That's not very helpful, but I don't really know.

Pieter Abbeel

The following question, here, also from Bishal. What's the next big thing on AI and advice for PhD students to which research area to focus on?

Geoff Hinton

I think a next big thing, I don't think there is the next big thing, a next big thing is going to be a convincing learning algorithm for spiking neural nets. That are able to deal with the discrete decision about whether to spike or not the continuous decision about exactly when to spike. And that makes use of spiked timing to do interesting computations that would be much more difficult to do in non spiking neural nets. That would be my bet about one of the big things. But the other thing and what the reason the deep learning revolution is going to keep going is that actually, if you just make a bigger one, you don't need any new ideas, you already get things working better. It's slightly depressing if your trade is new ideas, but if your trade is how do you build hardware to make a bigger one then it's great.

Pieter Abbeel

The next one is from thinkorswim. What is Professor Hinton's regret in research choices so far, that is something he wished he had delved into, but chose not to, and now perhaps a regret looking back?

Geoff Hinton

Time is short. So I'll just say a learning algorithm for spiking neural nets.

Pieter Abbeel

You wish you'd already done it, but now you can still do it in the next year.

Geoff Hinton

Yeah. Maybe.

Pieter Abbeel

Yordan Hristov has the following question, how important is embodiment for intelligence given the recent DALL·E results from OpenAI? And I'll say I'm personally really curious about that too, working on a lot of embodied intelligence myself.

Geoff Hinton

Yeah. So I think one needs to distinguish the engineering version of this question from the philosophical version of this question. So the philosophical version is, could a being sitting in a room listening to radio and watching a television figure out how the world worked, even if it couldn't actually move anything? It just gets these sensory inputs. And that's a philosophical question. I think it could. The engineering question is, is that a good approach just to listen to the radio and watch television? And I think the answer is definitely no. If you want to do perception, for example, as soon as you put one or two cameras on a robot and let the robot move around in the world, you get a very different view of what the questions are and how to solve them. Then, if your idea of doing perception is just to take a database of images like ImageNet because you have the option of changing viewpoints and seeing how these move and change viewpoint. You have a task to do. You have to be able to ignore things that aren't relevant. You really would like to have a fovea so you can see fine detail without swamping yourself all the time. It completely changes how you build your perception system. So philosophically, you don't need to be embodied, but actually as soon as you're embodied in a sensible way, it changes how you're going to do things. So for engineering embodiment is important. However, there's a lot of hassle that comes with embodiment. You have to deal with the body. So I think we can still make lots of progress on databases of just videos where there was somebody making the video, but basically you're just taking the video as data. There's lots of room for working like that without having a mobile robot. We don't control the data collection, but a long time ago, Dana Ballard, probably back in the eighties, Dana Ballard realized that animate perception when you've got a robot moving around is just going to have a very different flavor from standard computer vision. I think he was completely right.

Pieter Abbeel

Next one is from Renjith Ravindran. Why do you do what you do? Do you believe it would make the world a better place? Or are you just having fun exploring the limits of human creativity?

Geoff Hinton

Much more the second one, I'm afraid. So I really want to understand how the brain works. And I believe that to understand it, we need some new ideas. Like, for example, a learning algorithm of spiking neural nets.

Pieter Abbeel

Do you think it's, this is a follow up question of my own, you think it's almost necessary to be really driven by the exploratory aspects? Or is it possible to be just as productive in research if you care more about the

bottom line effect on the world? Is it just a different style?

Geoff Hinton

I think if you want to do fundamental research, there has to be curiosity. You're going to do your best research when it's curiosity driven. You're going to be motivated to ignore all the apparent barriers and pretend they're not there and see where you get. Whereas if it's for the bottom line, I just don't think you're going to be as creative. So I think the sort of the very best research gets done by graduate students in good groups with plenty of resources. So you need to be young and driven and really be interested in something.

Pieter Abbeel

Next one is from Peter Chen, actually my co-founder and CEO at Covariant, you know him. He has a research organization question. You've been doing pure academic, basic research at the university. You've done industry basic research at Google Brain, and you've also seen industry applied research while at Google, as well as some people you know, who are involved in startups and so forth. How do you think of these different places as providing, maybe, different opportunities to make research go forward, but also from there built products?

Geoff Hinton

To be honest, I don't think that much about building products. Products are nice. They pay the bills and companies would like to have products. It's not what I really care about. What I really care about is how do you make big learning systems and how does the brain work. And the nice thing about the brain team at Google is they have the resources to explore big systems and lots of smart people to discuss things with. And maybe I should care more about products, but I believe in specialization. And so having everybody care about products is not necessarily the right mix.

Pieter Abbeel

The next batch of questions is all centered around the brain. So I'm going to give you all the questions in one go, Geoff. And then you can see what perspective you want to give on this whole thing. The first one from Lucas Beyer is how does the brain work? Then Tim Dettmers, what's your take on mixed learning algorithms: backprop in cell body dendrites plus feedback alignment across neurons? Could such algorithms be both biologically plausible and competitive with pure backprop, or is a single general algorithm more likely to exist? Prasad Kothari is wondering about spiking neural networks. Cedric Vandelaer, it seems you have drawn inspiration from the human brain in the past. Do you think there are certain techniques that will eventually turn out to be crucial? For example, spiking neural networks. at and commander. Aton Kamanda, Geoff recently declared that he finally didn't think the brain was doing backpropagation, but it might be doing something akin to the Boltzmann machines. Does he see this kind of architecture comeback as a viable AI model or as a theoretical model for how the brain works? And then the last one by Yigit is about the NGRAD hypothesis. So it's a lot of related questions here.

Geoff Hinton

There's one set of issues, which is if the brain is going to do something like backprop, how does it get gradient information to go backwards through the layers. And that's what the NGRAD hypothesis is about. It is the idea of using changes in your neural activity, to represent error derivatives, using temple derivatives for error derivatives. I didn't really believe in that anymore. So let me go to the question of Boltzmann

machines and do I believe in both machines? I wax and wane on Boltzmann machines because it's such a neat idea. But right now I believe in part of that, but not the main thing. So Boltzmann machines have these Markov chains which require symmetric weights, which are implausible. But there's another aspect of Boltzmann machines that I mentioned in the podcast, which is that they use contrastive learning so Boltzmann machines are more like a GAN than like typical unsupervised contrastive learning. In an unsupervised contrastive learning, you take a pair of crops of the same image and make their representations similar and look at the pair of crops of two different images and say make that representation not dissimilar. In a Boltzmann machine, you take positive data and say, have low energy for the positive data. You take negative data and you say, have high energy for the negative data. So, but the data is just an image, it's not a pair of images. It's just an image. And I believe in that now. So I think that if we're going to get unsupervised contrastive learning working, what we need is to have two phases like in a Boltzmann machine. We can have a phase when you try to find structure in positive data. But not in pairs, the whole image. You're looking around for, essentially agreements between locally extracting things that detection predicted things. And then we need a different phase in which I show you negative images, things like real images that aren't real or slightly different. And what you're concerned with is that the structure you find in real images shouldn't be in these negative images. So you want to find things that are in the positive data and not in the negative data. And that's how you protect yourself from finding structure inside your neural network. This is caused by the wiring of the front end of the neural network. Anything caused by the wiring will cause the same structure for positive images and negative images. And so you can filter it out. So there's an aspect of Boltzmann machines I really believe in, which is you have to use positive and negative data to protect yourself from just learning about your own wiring. But the idea of a Markov chain to generate the negative data, I think, is just too cumbersome. I think we need other ways of generating the data and this is quite like GANs, right? So in GANs you've got real data and you've got data generated by generative models and that's negative data. And if you compare what I believe now with GANs, what I believe is that the discriminator, which is trying to decide is this real or negative data. And by finding structure, it should only be there if it is real data. That's the sort of main thing. And I want to use the internal representations of the discriminator as a generative model, in order to get the negative examples for training. So what I'm doing is kind of, what I believe in now is sort of a cross between GANs and Boltzmann machines. But in GANs, it's not a Markov chain, it's generative models just a causal generative model, a directive generative model which is much easier. And I think you probably have a discriminator. And then the directed generative model that's around at the same time for the negative examples.

Pieter Abbeel

In principle, there is a unification right? Because GANs can be rewritten as energy based models. Also just a specific form of them.

Geoff Hinton

But the thing about GANs is you generate from random stuff at the top. And it's hard to get coverage. There might be all sorts of things you never generate. You wouldn't know. If your discriminator, you go to the top level of your discriminator, and then you regenerate from the top level of your discriminator, you'll get coverage. So in a paper with the wake-sleep algorithm that I published in 2006 with Simon Osindero and Yee-Whye Teh,on neural computation, we have something that doesn't use backprop. It manages without backprop. It uses contrastive wake-sleep. And the contrasting aspect is that you do recognition, that's the sort of wake phase and then you generate. And what you generate from is not random stuff, but a

perturbation of what you got when you did recognition. And that gives you coverage. So that's I think there's maybe unification coming on.

Pieter Abbeel

That seems a very concrete idea, ripe for execution and could get some amazing results.

Geoff Hinton

It's actually running on my computer right now. It is running on my computer right now. Oh, you're.

Pieter Abbeel

Oh you're running it right now. Got it. And then the other batch of questions related to the brain was, of course, on spiking, the role of spiking.

Geoff Hinton

Well, I think it's very important. I think from very early on in neural nets, Minsky and Papert they hit on the XOR as the thing that a neuron couldn't do, right? They couldn't tell whether two inputs were different. It's an exactly equivalent problem to solve the same function. You can't tell whether the two inputs are the same. Obviously, if you could do one you could do the other. It's unfortunate that they went for XOR rather than the same. Because if you go for same and say well, I'll artificially neurons can't tell if our two inputs are the same, you're immediately drawn to the idea that, well, if you use spike time, you can tell whether two spikes arrived at the same time, because then they push a lot of charge into the neuron at the same time. And we will put it above threshold, particularly if the excitatory inputs followed by some inhibitory inputs. So they have to arrive in a narrow window. So spiking neural networks are very good at detecting agreement and our normal neural networks need several layers to do that. And if we could just get a good learning algorithm, I think we would discover that they learn to make use of that ability, just like they learn to make use of it really well for doing auditory localization.

Pieter Abbeel

When I think about the transformer architectures. They're also kind of designed to find agreements or correlations. Just a much more, I guess, much bigger piece of machinery than maybe a spiking architecture. But it seems like there could be some connections there.

Geoff Hinton

I mean, there've been neuroscientists saying for years and years that it'd be crazy not to use the spike time. And there are people like appearance and talked about seeing functions. Um. It would be very satisfying to find a learning algorithm for these things and show that when you start learning, particularly on sequential data like auditory data, then they really do make use of the spike times in a sensible way. And then you could use the spiking cameras. Spiking cameras are very clever things that give you lots of information, but nobody knows how to use it. Same with your auditory domain, people like Dick Lyon have been saying for years, we should be using spiking neural nets to represent auditory input, but nobody knows how to then take that representation and learn on it and do things with it.

Pieter Abbeel

There's a follow up question of my own, but if I think about spiking and let's say I try to play devil's advocate here and try to maybe argue against a strong belief in spiking, I might say something along the lines of, well, maybe the reason we have spiking in human brains is because maybe evolutionarily it was easier to somehow evolve or due to random luck of the draw, we evolved spikes. But we didn't evolve wheels and wheels are maybe more effective at transportation.

Geoff Hinton

Oh no, we did evolve them. You do have wheels.

Pieter Abbeel

Oh, I do?

Geoff Hinton

You do have wheels. You just need to think straight. So you have to go over rough ground, right? So you need a wheel with a six foot diameter. And that's going to be a lot of rim. Okay. So as soon as you know about timesharing, you decide, well, here's what I'm going to do. I'm going to have a wheel with a six foot diameter but I'm actually only going to have two little bits of the rim. And I'm going to alternate between using these two bits of the rim and I'm going to use it as a wheel. So I'm going to rotate about the hip, which is going to be a very low energy way of walking. And then I'm suddenly going to switch because I have to get this to go backwards. I have to fly back, all the way round. I'm going to fly back and then I'm going to use the other leg, the other bit of rim. And there is one other big difference, which is a normal wheel, the axle is suspended from the top of the wheel. And there's pressure in the side of the wheel to hold it up so the spokes are in tension. You have to have something like the rubber tire for rough ground. So what you have is you, instead of spoke that's in tension, you have a spoke that's in compression. You just have one of them for each bit of rim. But it can bend in the middle. That means you don't need tire because you can absorb a lot that way. And you don't have too much unsprung weight because you can get a bit of the rim. But it's basically a wheel. It's just a time-shared wheel. And now there is one other little advantage the time-shared wheel has, which is you don't have a problem in getting nutrients in because it doesn't go all the way around. It just goes forwards and backwards. So you can have blood vessels go to it more easily. But mechanically that's just an energy supply problem, mechanically, it really is a wheel. You're using it just like a wheel, a little bit of rim and you're rolling like a wheel does. So you use very little energy. And then you quickly substitute one piece of rim for the other. I'm surprised you didn't know we had wheels.

Pieter Abbeel

So maybe my bad analogy aside, Geoff, do you think there's any possibility that just biking was easier to evolve and that's why we ended up with?

Geoff Hinton

No, I think no, I think there's a very good reason for using it.

Pieter Abbeel

Got it.

Geoff Hinton

But I don't know what it is. I think it's to do with coincidence detection. Next thing you'll be saying is that when we make flying machines, we don't give them feathers.

Pieter Abbeel

Well, I wasn't going to go there after you so eloquently had told me I have a wheel.

Geoff Hinton

That's what's wrong with drones, right? If you have a drone and the blade hits something, either it breaks a thing or it breaks the blade. If the blade was made of little bits of velcro that zipped together, when it hit something, it could break. And then the drone would land and it'd do a bit of preening and zip the velcro back together again and it could fly off again. So really we ought to make drones with feathers instead of rigid blades so that they could hit things without damaging them and without damaging themselves. And have something that would preen the feathers back together again and off we go. So those are the two classic examples. People don't have wheels and airplanes don't have feathers. Well, they're both wrong. Drones don't have yet, but I think they will.

Pieter Abbeel

I'll be interested to see when that happens. So the next couple of questions are again, quite related so I'm going to ask them in batch. Abdullah Hamdi asked, what's the next paradigm shift in AI after deep learning? Ajay Divakaran, does the current deep learning paradigm suffice for transfer learning like humans? Or does it need to be fundamentally enhanced? And Arun Rao, what are the next milestones for deep learning going from existing foundation models to a long term goal of AGI? And how does Hinton define AGI?

Geoff Hinton

I try to avoid defining AGI and I try to avoid working on AGI because I think AGI, there's all sorts of things wrong with the vision of AGI. It envisions an intelligent human like Android, that's as smart as us, and I don't think intelligent is necessarily going to develop like that. I think I'm hoping it develops more symbiotically. It is very individualistic and we developed in communities. We develop, so this goes back to what you said in the podcast about ants and so on, I think intelligence develops in societies better than it does individualistically. And I think maybe we'll get smart computers, but they won't be autonomous in the same way. They may have to be if they're for killing other people. But hopefully that's not where we are going.

Pieter Abbeel

Yeah. The earlier part was about the next transition? What's next after deep learning? I mean, that's the question. I'm not trying to apply that there is something next, but that's the question.

Geoff Hinton

Right. So what I believe is this, that we won't, we're going to stay with the very successful paradigm of tuning a lot of real life parameters based on the gradient to some objective function. I think we'll stay with that. But we may well not be using backpropagation to get the gradient and the objective functions may be far more local and distributed. That's where I think we are headed.

Pieter Abbeel

Next question is some Dystopia Robotics. Are you familiar with Richard Sutton's The Bitter Lesson?

Geoff Hinton

Oh, yes.

Pieter Abbeel

And what are your thoughts on it?

Geoff Hinton

I sort of have it in my lectures that deep learning depends on two things. It depends on doing stochastic gradient descent in big networks that have a lot of data and a lot of compute power. And then on top of that, there's a few ideas that make it work a little bit better. Things like dropout and all the stuff we've worked on, all make it work a little bit better. But the crucial thing is lots of compute power, lots of data and stochastic gradient descent. And I agree with it.

Pieter Abbeel

Next question is from Prabhav Kaula. How do you read research papers? How to get past the mathematics and get a taste of the core message?

Geoff Hinton

Okay. I don't read many research papers. I basically get my colleagues and my students to explain them to me. I'm hopeless at mathematics. I can do what I have to, to justify something I've already thought out. Like Boltzmann machines, I've figured out how they would work and then did the math to show that's the right thing to do. But I'm not very good at math, and I always find it a big barrier reading papers to understand all the notation. And I find it much easier if I get, so for neuroscience, I get Terry Sinofsky to explain it to me. And for computer science, I get my grad students to explain to me.

Pieter Abbeel

Very related question to what you just answered, Geoff, is from Chaitanya Joshi. Many people have shared anecdotes on how Professor Hinton's mind works in an analogical and intuitive manner, with an aversion to mathematics and proofs. Could Prof. Hinton elaborate on the roles of formalism versus intuition when going about research?

Geoff Hinton

I think there's room for more than one kind of person, so I sort of hate formalism. I love intuition. I love tinkering about on my Mac to see what works and what doesn't. I think it's very important to have foundational work to really understand the mathematical foundations of things. It's not what I do. It's good to have proofs. It's not what I do. I have a little test I give people. Suppose there were two talks going on at NIPS, at the same time and you had to decide which one to go to. One talk was about a really clever and elegant way, a new, totally new way of proving unknown results. And the other talk was about a new learning algorithm that seems to do amazing things, but nobody understands why. Now I know which talk I'd go to. And I know that the, it was easier to get the first paper accepted than the second one. But other people would really like to know new ways of proving things because that's what they think is really interesting. I'm not like that at all. But I actually think nearly all the progress in neural nets has not come from doing the math right. It's come from intuitive ideas. But later on, people do the math.

Pieter Abbeel

That definitely resonates with me. Guillermo Martinez Villar asks how do you transition from a background in psychology to the field of AI? And what would you suggest to young people considering doing the same?

Geoff Hinton

Okay so there is an interesting issue there. So when I was teaching at the U of T, if you looked at the undergraduates, there were a lot of computer science undergraduates who are very good. They're also cognitive science undergraduates who did minor work in computer science, but really cognitive scientists. And they typically weren't quite as good at the technical stuff, but they've gone on to do much better things because they had the interest in the issues. They really wanted to understand how cognition worked. So I'm thinking of people like Blake Richards and Tim Lillicrap, who've gone on to do great things. Because they knew what questions they wanted answered, whereas most computer scientists didn't. And for some reason I thought that was relevant to the question. Could you say the question again?

Pieter Abbeel

Oh, yeah, it is very relevant to the question. Let me tee it up again. How did you transition from a background in psychology to the field of AI? And what would you suggest to young people considering doing the same?

Geoff Hinton

I don't know. It's very hard to generalize. I had a very weird career where I started off doing physics and physiology in my first year at university. In fact I was the only student at Cambridge that year doing both physics and physiology. And then my math wasn't good enough for physics. And I wanted to know the meaning of life so I did philosophy. And developed strong antibodies. I'm going to do psychology. But I did have some quantitative background having done physics and physiology. So retrospectively, it was an interesting background. It didn't happen with any design. It just kind of happened. But I think you need to have questions that you're driven by and not just techniques. It's more important to have questions that really excite you and you'd do anything to find the answer than to just be very good at some technique. However, I wish I learned more from math when I was young. I wish I didn't find linear algebra complicated.

Pieter Abbeel

Next question is from Khalid Saifullah. How conscious do you think, if at all, are today's neural networks?

Geoff Hinton

I guess Ilya would say just a little bit and get lots of flak for saying that. I have a view about consciousness. So about a hundred years ago, if you ask people what distinguishes living things from dead things, they'd say, well, living things have vital force and dead things don't. And if you say what's vital force? They'd say, well, it is what living things have. And then we developed biochemistry and we understood about how biochemical processes work. But since some people haven't talked about vital force, it's not that we don't have vital force. We still have vital force if we ever had it. It's just not a useful concept anymore because we understand in detail how things work at the biochemical level, and we understand that the organs break down when they don't get enough oxygen, and then you're dead, and then it all decays. And it's not like some vital force left the body and went to heaven, it's that the biochemistry just packed up on you. So I think the same is going to be true of consciousness. I think consciousness is a pre-scientific concept and I think

that's why people are very bad at defining it and everybody disagrees. And I don't have any use for it. There's there's many related concepts like, are you aware of what's going on in your surroundings? So if Muhammad Ali hits you on the chin, you're not aware of what's going on in your surroundings. And we use the word unconscious for that. That's one meaning of unconscious. But if I'm driving without thinking about what I'm doing, that's another meaning of unconscious. We have all these different meanings. And my view of consciousness is it's a kind of primitive attempt to understand us, to deal with what's going on in the mind by giving it a name and assuming it the sum essence that explains everything. And here's a similar analogy for cars. If you don't understand much about cars, I can tell you how cars work. Cars have oomph, and some cars have more oomph than others. Like one of these Teslas with big batteries has a lot of oomph and a little mini doesn't have much, especially if it's old. And that's how cars work. Because some have more oomph than others. And obviously, if you want to understand cars, it's really important to understand oomph. Now, as soon as you get down to understanding oomph, you start understanding how engines work and torque and energy and how it's converted and all that stuff. But as soon as you start understanding that, you stop using the word oomph. And I think it's going to be us for consciousness.

Pieter Abbeel

I love this explanation, Geoff. Farzana Mary's Azad asks, ML once started with roots in human psychology. Do you see ML advancements, today, having the capacity to help better understand human psychology in the future? Like seeing people as neural networks or classifiers and their cognitive distortions similar to under-overfitting and so forth.

Geoff Hinton

Yes, I do. I strongly believe that. I strongly believe that when we eventually understand how the brain works, that's going to give us lots of psychological insight too. Just as understanding chemistry at the atomic level, understanding of how molecules bump into each other and what happens, gives us lots of insight into the gas laws. The fine level understanding is important and does give rise to understanding what's going on at high levels. And I think it's going to be very hard to get satisfactory explanations of a lot of things for high levels. Things like schizophrenia, for example, without understanding the details of how it all works.

Pieter Abbeel

Well, thank you, Geoff, for making the time for the additional Q&A section with questions from our audience. Wow. What a way to wrap up season two. Thanks so much for all the great questions for Geoff. And thanks for listening to the podcast. If you enjoy the show, please consider giving us a rating. And please recommend us to your friends and colleagues.