**CRAIG:** I wanted to have you explain to listeners forward-forward networks and why you're looking for something beyond backpropagation, despite its tremendous success.

**GEOFF:** Let me start with explaining why I don't believe the brain is doing back propagation. One thing about back propagation is you need to have a perfect model of the forward system that is in backpropagation.

**GEOFF:** It's easiest to think about for a layered net, but it also works for recurrent nets. For a layered net, you do a forward pass where the input comes in at bottom and goes through these layers. So, the input might be pixels and what comes out the top might be a classification of is it a cat or a dog? You go forwards through the layers and then you look at the error in the output.

**GEOFF:** If it says cat when it should say dog, that's wrong. And you'd like to figure out how to change all the weights in the forward pass so that next time it's more likely to say the right category rather than the wrong one. So, you have to figure out how a change in a weight would affect how much it gives the right answer.

**GEOFF:** And then you want to go off and change all the weights in proportion to how much they're helping getting the right answer. And back propagation is a way of figuring out that gradient, a way of figuring out how much a change in the weight would make the system have less error. and then you change the weight in proportion to how much it helps.

**GEOFF:** And obviously if it hurts you change it in the opposite direction. Now backpropagation looks like the forward pass, but it goes backwards. It has to use the same connectivity pattern with the same weights, but in the backwards direction. And it has to go backwards through the non-linearity of the neuron.

**GEOFF:** There’s no evidence that the brain is doing that. And there's lots of evidence it's not doing that. So, the worst case is if you're doing back propagation in a recurrent net, because then you run the recurrent net forwards in time, and it outputs an answer at the end of running forwards in time.

**GEOFF:** And then you have to run it backwards through time in order to get all these derivatives. So, I had to change the weights and that's particularly problematic if, for example, you're trying to process video. You can't stop and go backwards in time. So combined with the fact that there's no evidence the brain does it - or no good evidence - there's the problem that just for technology it's a mess.

**GEOFF:** It interrupts the pipelining of stuff through. So, you'd really like something like video, there's multiple stages of processing, and you'd like to just pipeline the input through those multiple stages and just keep pipelining it through. And so, the idea of the forward algorithm is that if you can divide the learning, the process of getting the gradients you need, into two separate phases, you can do one of them online and one of them offline.

**GEOFF:** And the one you do online can be very simple and will allow you to just pipeline stuff through. So, the online phase, which is meant to correspond to wake, you put input into the network and, let's take the recurrent version, input keeps coming into the network.

**GEOFF:** And what you're trying to do for each layer at each time step, you're trying to make the layer have high activity. Or rather high enough activity so that it can figure out that this is real data. So, the underlying idea is, for real data, you want every layer to have high activity, and for fake data - and I'll come to how we get that later -you'd like every layer to have low activity. And the task of the network, or the thing it's trying to achieve, is not to give the correct label, as in back propagation. It's trying to achieve this property of being able to tell the difference between real data and fake data at every layer, by each layer having high activity for real data and low activity for fake data.

**GEOFF:** So, each layer has its own objective function. In fact, to be more precise, we take the sum of the squares of the activities of the units in a layer, we subtract off some threshold, and then we feed that to a logistic function that simply decides what's the probability that this is real data as opposed to fake data.

**GEOFF:** And if the logistic function gets a lot of input, it'll say it's definitely real data. And so, there's no need to change anything if it's getting lots of input. It won't learn on that example because it's already getting it right. And that explains how you can run lots of positive examples without running any negative examples, which are fake data because it'll just saturate on positive examples that it is getting right. So that's what it does in the positive phase. It tries to get high sum of squared activities in every layer high enough so that it can tell that it's real data.

**GEOFF:** In the negative phase, which is run offline, that is during sleep, the network needs to generate its own data and, given its own data as input, it wants to have low activity in every layer. So, the network has to learn a generative model. And what it's trying to do is discriminate between real data and fake data produced by its generative model.

**GEOFF:** Obviously, if it can't discriminate at all, then what's going to happen is the derivatives that it gets for real data and the derivatives that it gets for fake data will be equal and opposite. So, it won't learn anything. Learning will finish then if it can't tell the difference between what it generates and real data.

**GEOFF:** This is very like a GAN if you know about generative adversarial networks. Except that the discriminative net is trying to tell the difference between real and fake, and the generative model that's trying to generate fake data use the same hidden units. And so, they use the same hidden representations. That overcomes a lot of the problems that a GAN has.

**GEOFF:** On the other hand, because it's not doing back propagation to learn, the generative model is harder to learn a good generative model. That's a rough overview of the algorithm.

**CRAIG:** Let me ask a couple of questions on the wake and sleep cycle. Are you cycling quickly between them?

**GEOFF:** Okay. So, most of the research, what I would do is - the preliminary research cycle - quickly cycle between them because that's the obvious thing to do. And later on, I discovered, I've known for some time that with contrastive learning, you can separate the phases. And later on, I discovered it worked pretty well to separate the phases.

**GEOFF:** Recent experiments I've done with predicting characters, you can have it predict about a quarter of a million characters.

**GEOFF:** So, it's running on real data, trying to predict the next character. It's making predictions, it's running with mini batches. So, after making quite a large number of predictions, it updates the weights and then it sees more positive examples. It updates the weights again. So, in all those phases, it's just trying to get higher activity in the hidden layers - but only if it's not already got high activity.

**GEOFF:** And you can predict like quarter of a million characters in the positive phase. And then switch to the negative phase where the network's generating its own string of characters. And you are now trying to get low activity in the hidden layers for the characters.

**GEOFF:** It's predicting, it's looking at a little window of characters.

**GEOFF:** And then you run for quarter of a million characters like that. And it doesn't actually have to be the same number anymore. With Boltzmann machines, it's very important to have the same number of things in the positive phase and negative phase, but with this it isn't.

**GEOFF:** And what's remarkable is that up to a few thousand predictions, it works almost as well if you separate the phases as opposed to interleaving. And that's quite surprising.

**CRAIG:** In human learning, certainly there's wake and sleep for complicated concepts that you're learning, but there's learning going on all the time that doesn't require a sleep phase.

**GEOFF:** There is in this, too, if you're just running on positive data. It does a lot of learning in the positive phase. But if you go on too long, it fails catastrophically, and people seem to be the same.

**GEOFF:** If I deprive you of sleep for a week, you'll go completely psychotic, and you may never recover.

**CRAIG:** I think one thing that non practitioners are having trouble understanding is the concept of negative data. I've seen a few articles where they just put it in quotation marks out of your paper, which indicates that they don't understand it.

**GEOFF:** Okay. What I mean by negative data is data that you give to the system when it's running in the negative phase. That is when it's trying to get low activity in all the hidden layers. And there are many ways of generating negative data. In the end, you'd like the model itself to generate the negative data.

**GEOFF:** So, this is just like it was in Boltzmann machines. The data that the model itself generates is a negative teacher and real data is what you're trying to model. And once you've got a really good model, the negative data looks just like the real data, so no learning takes place. But negative data doesn't have to be produced by the model.

**GEOFF:** So, for example, you can train it to do supervised learning by inputting both an image and the label. So now the label's part of the input, not part of the output. And what you're asking it to do is, when I input an image with the correct label, that's going to be the positive data and you want to have high activity.

**GEOFF:** And when I input an image with the incorrect label, which I just put in by hand as an incorrect label, that's negative data. Now it works best if you get the model to predict the label and you put in the, best of the model's predictions that are not correct, because then you're giving it the things it's most to make a mistake with, it's most likely to make as negative data. But you can put in negative data by hand, and it works fine.

**CRAIG:** And the reconciliation then at the end, is it, as in Boltzmann machines, where you're subtracting the negative data from the positive data?

**GEOFF:** In Boltzmann machines what you do is you give it positive data, real data, and you let it settle to equilibrium, which you don't have to do with the forward-forward algorithm.

**GEOFF:** Or not exactly anyway. And once it's settled to equilibrium, you measure the pairwise statistics. That is how often two units that are connected are on together. And then in the negative phase you do the same thing You just let the model settle as producing data itself. And you measure the same statistics, and you take the difference of those pairwise statistics.

**GEOFF:** And that is the correct learning signal for a Boltzmann machine. But the problem is you have to let the model settle. And there just isn't time for that. Also, you have to have all sorts of other conditions, like the connections have to be symmetric. There's no evidence connections in the brain are symmetric.

**CRAIG:** Can you give a concrete example of positive and negative data in a very simple learning exercise? You were working on digits.

**GEOFF:** The simplest example, I think, is if you're predicting a string of characters. For the positive data, you'd see a little window of characters and you'd have some hidden layers.

**GEOFF:** And because that's a positive window of characters, you'd try and make the activity high in all the hidden layers. But also from those hidden layers, from the activity in those hidden layers, you would try to predict the next character. So that's a very simple generative model. But notice the generative model isn't having to learn its own representations.

**GEOFF:** The representations are learned just to make positive strings of characters give you high activity in all the hidden layers. That's the objective of the learning. The objective isn't to predict the next character. But having done that learning, and gotten the right representations for these strings of characters, these windows of characters, you also learn to predict the next character.

**GEOFF:** And that's what you're doing in the positive phase. So, you're seeing windows of characters, you're changing the weights so that all the hidden layers have high activity for those windows of characters. But you're also changing top-down weights that are trying to predict the next character from the activity in the hidden layers.

**GEOFF:** That's what's sometimes called a linear classifier. So that's a positive phase. In the negative phase, as input, you use characters that have been predicted already. So, you've got this window and you're going along and just predicting the next character, and then moving the window along one to include the next character you predicted and to drop off the oldest character.

**GEOFF:** And you just keep going like that. And for each of those frames you try and get low activity in the hidden layers because it's negative data. And I think you can see that if your predictions were perfect and you start from a string, a real string, then what's happening in the negative phase will be exactly like what's happening in the positive phase.

**GEOFF:** Right? And so, the two will cancel out. But if there's a difference, then you'll be learning to make things more like the positive phase and less like the negative phase. And so, it'll get better and better at predicting.

**CRAIG:** As I understand back propagation, on static data there are inputs, there's an output and you calculate the error, and then you run backward through the network and correct the weights, and then do it again. And that's not a good model for the brain because there's no evidence of information flowing backward through the neurons.

**GEOFF:** That's not exactly the right way to say it. There's no, no good evidence of derivative information flowing backwards; that is, these error gradients flowing backwards. Obviously, the brain has top-down connections. If you look at the perceptual systems, there's a kind of forward direction that goes from the thalamus up to temporal cortex where you recognize things.

**GEOFF:** And the thalamus is where the input comes in from the eyes. And there's connections in the backward direction, but the connections in the backward direction don't look at all like what you'd need for back propagation. For example, in two cortical areas, the connections coming back don't go to the same cells as connections going forward come from. It's not reciprocal in that sense.

**GEOFF:** There's a loop between the cortical areas. But information in one cortical area goes through about six different neurons before it gets back to where it started. And so, it's a loop. It's not like a mirror system.

**CRAIG:** Okay. But my question is, you talk about turning the static image into a 'boring' video.

**GEOFF:** That allows you to have top-down effects. Think of it being a forward direction, which is going from lower layers to higher layers, and then orthogonal to that is the time dimension. And so, if I have a video, even if it's a video of just a single thing that stays still, I can be going up and down through the layers as I go forwards in time.

**GEOFF:** And that's what's allowing you to have top-down effects.

**CRAIG:** Yeah. Each layer receives inputs from a higher layer in the previous time step.

**GEOFF:** Exactly. Yeah. So, what a layer is doing is receiving input from higher layers and lower layers at the previous time step and from itself at the previous time step.

**GEOFF:** And if you've got static input, that whole process over time looks like a network settling down. That's a bit more like a Boltzmann machine settling down. And the idea is that the time that you're using for that is the same as the time you're using for posting video. And because of that, if I give you input that's changing too fast, you can never settle down to interpret it.

**GEOFF:** So, I discovered this nice phenomenon: if you take an irregularly shaped object, like a potato, for example, a nice irregularly shaped potato, and you throw it up in the air rotating slowly at one or two revolutions per second, you cannot see what shape it is. You just can't see the shape of it. You don't have time to settle on a 3D interpretation.

**GEOFF:** Because it's the very same time steps that you're using for posting videos you're using for settling with a static image.

**CRAIG:** What I found fascinating and, maybe this is something that is already in the literature, but this idea of, going up and down in the layers as you move through time.

**GEOFF:** That's always been in recurrent nets.

**GEOFF:** So, to begin with, recurrent nets, we just have one hidden layer. So typical LSTMs and so on would have one hidden layer. And then Alex Graves pioneered the idea of having multiple hidden layers and showed that it was a winner. So that idea's been around, but it's always been paired with back propagation as the learning algorithm.

**GEOFF:** And in that case it was back propagation through time, which was completely unrealistic.

**CRAIG:** And real life is not static, so you're not perceiving in a truly static fashion. How much of this grew out of SimCLR's contrastive learning or activity differences.

**GEOFF:** A couple of years ago I got very excited because I was trying to make a more biologically plausible version of things like SimCLR. There's a whole bunch of things like SimCLR. SimCLR wasn't the first of them. In fact, there's something a bit like SimCLR that Sue Becker and I published in 1992 in Nature.

**GEOFF:** But we didn't use negative examples. We tried to analytically compute the negative phase, and that was a mistake. It just, that would never work. Once you start using negative examples, then you get things like SimCLR. And I discovered that you could separate the phases, and that got me very excited a few years ago because it seemed like I finally had an explanation for what sleep was for.

**GEOFF:** One big difference, SimCLR is taking two different patches from the same image. And if they're from the same image, it's trying to make them have a similar representation. If they're from different images, it's trying to make them have different representations, sufficiently different - once they're different, it doesn't try and make them more different.

**GEOFF:** And SimCLR involves looking at two representations and seeing how similar they are. And that's one way to measure agreement. And in fact, if you think about the square difference between two vectors, that decomposes into three terms. There's something to do with the square of the first vector.

**GEOFF:** There's something to do with the square of the second vector, and then there's the scalar product of the two vectors. And the scalar product of the two vectors is the only interactive term. And so, it turns out that square difference is very like a scalar product. A big square difference is like a small scalar product.

**GEOFF:** Now there's a different way to measure agreement, which is to take the things you'd like to agree and feed them into one set of neurons. And now if two sources coming into that set of neurons agree, you'll get high activity in those neurons. It's like positive interference between light waves. And if they disagree, you'll get low activity.

**GEOFF:** And if you measure agreement just by the activity in a layer of neurons, you're measuring agreement between the inputs, then you don't have to have two things. You can have as many things as you like. You don't have to divide the input into two patches and say, do the representations of these two patches agree.

**GEOFF:** You can just say, I've got a hidden layer. Does this hidden layer get highly active? And it seems to me that's a better way to measure agreement. It's easier for the brain to do. And it's particularly interesting if you have spiking neurons because what I'm using at present doesn't use spiking neurons.

**GEOFF:** It just says a hidden layer is really asking, are my inputs agreeing with each other? In which case they'll be highly active. Or are they disagreeing? In which case they won't. But if the inputs arrive at specific times, very precise times like spikes do, then you can ask, not just are the stained neurons being stimulated, but are they being stimulated at exactly the same time?

**GEOFF:** And that's a much sharper way to measure agreement. So spiking neurons seem particularly good for measuring agreement, which is what I need. That's the objective function, to get agreement in the positive phase and not in the negative phase. And I'm thinking about ways of trying to use spiking neurons to make this work better. But that's one big difference from SimCLR, that you're not taking two things and saying, do they agree?

**GEOFF:** You're just taking all the inputs coming into a layer and saying do all those inputs agree?

**CRAIG:** When you talk about the activity, that's similar to what you were doing with N grads where you're comparing top-down predictions and bottom-up predictions.

**GEOFF:** Okay. When you do the recurrent version of the forward-forward algorithm at each time step neurons in a layer are getting top-down input and bottom-up input, right?

**GEOFF:** And they'd like them to agree. And if your objective function is to have high activity, they'd like to make things highly active. There's another version of the forward-forward algorithm where the objective is to have low activity and then you want the top down to cancel at the bottom up. And then it looks much more like predictive coding.

**GEOFF:** It's not quite the same, but it's very similar. But let's stick with the version where you're going for high activity. You want the top down and bottom up to agree and give you high activity. But notice that it's not like the top down is a derivative. So, in attempts to implement back prop in neural nets, you try and have top-down things which are like derivatives and bottom-up things, which are like activities.

**GEOFF:** And you try and use temporal differences to give you the derivatives and that's somewhat different. Here, everything's activities. You're never propagating derivatives.

**CRAIG:** And this algorithm also does away with the idea of dynamic routing that you talked about with stacked capsule encoders.

**GEOFF:** Yes. So, with capsules, I moved on from the dynamic routing to having what I called universal capsules. A capsule would be a small collection of neurons. And in the original capsules models, that collection of neurons would only be able to represent one type of thing, like a nose and a different kind of capsule would represent a mouth.

**GEOFF:** In universal capsules, what you'd have is that each capsule could represent any type of thing, so it would have different activity patterns to represent the different kinds of thing that might be there. The capsule would be dedicated to a location of the image, so a capsule would be representing what kind of thing you have at that location at a particular level of the part-whole hierarchy.

**GEOFF:** So, it might be representing that at the part level you have a nose and then at a higher level you'd have other capsules that are representing, at the object level, you have a face or something. But when you get rid of the dedication of a bunch of neuros to a particular type of thing, you don't need to do routing anymore.

**GEOFF:** And in the forward-forward algorithm I'm not doing routing. And one of the diagrams in the paper from the forward algorithm is actually taken from my paper on part-whole hierarchies, my last paper on capsule models. So, I had a system called GLOM, an imaginary system. And the problem with it was I never had a plausible learning algorithm.

**GEOFF:** And the forward algorithm is a plausible learning algorithm for GLOM. It's something that's nearly reasonable.

**CRAIG:** What was fascinating to me at least about capsules is that they captured the 3D nature of reality, right?

**GEOFF:** Lots of neural nets are now doing that. So, Nerf models, neural radiance field models, are now giving you very good 3D models in neural nets. So, you can see something from a few different viewpoints and then produce an image of what it would look like from a new viewpoint. That's very good, for example, making smooth videos from frames that are taken a quite long time intervals.

**CRAIG:** But in the forward-forward algorithm, what's your intuition? That, if indeed everything works out, that this is a model for information processing in the cerebral cortex and that perception of depth and the 3D nature of reality would emerge?

**GEOFF:** Yes. That's the hope. Yes.

**GEOFF:** In particular, if I'm showing you a video and the viewpoint is changing during the video, then what you'd want is that the hidden layers should represent 3D structures. That's all pie in the sky at present. I haven't gotten to that stage,

**CRAIG:** But with capsules, you referred to pixels having depth so that if one object moved in front of another the system understood that it was behind the thing in front of it. Do you capture that with forward-forward?

**GEOFF:** You would want it to learn to deal with that, yes. I wouldn't wire that in. But it's an obvious feature of video that it should learn about. With babies, they learn in just a few days to get structure from motion. That is, if I take a static scene and I move the observer, or if I keep the observer stationary and - the experiments were done with a piece of paper folded into a W and if you see it the wrong way around, it looks weird.

**GEOFF:** Experiments done by Elizabeth Spelke and other people use the idea that you can tell a lot about the perception of a baby by seeing what they're interested in because they're interested in things that look odd. And so, they'll pay more attention to things that look odd and within a few days they learn to deal with how 3D structure is related to motion.

**GEOFF:** And if you make it related wrong, they think it's weird. So, they learn that very fast, whereas it takes them like at least six months, I think, to learn to do stereo. To get it from the two eyes. It's just much easier to get from video than from stereo.

**GEOFF:** But from an evolutionary point of view, if something's really easy to learn, there's not much point wiring it in.

**CRAIG:** You've been working in MATLAB famously now on toy problems. Are you starting to scale? Are you still refining?

**GEOFF:** I'm doing a bit of scaling. I'm using a GPU to make things go a bit faster. But I'm still at the stage where there's very basic properties of the algorithm. I'm exploring, in particular, how to generate negative data effectively from the model. And until I've got the sort of basic stuff working nicely, I think it's silly to scale it up. As soon as you scale it up, it's slower to investigate changes in the basic algorithm. And I'm still at the stage where there's lots and lots of different things I want to investigate.

**GEOFF:** For example, here's just one little thing that I haven't had time to investigate yet: you can use as your objective function to have high activity in the positive phase and low activity in the negative phase. And if you do that, it'll find nice features in the hidden units. All you can have is your objective function to have low activity in the positive phase.

**GEOFF:** If you do that, it'll find nice constraints. If you think about what physicists do, they try and understand nature by finding apparently different things that add up to zero - another way of saying is that they're equal and opposite. But if you take force and you subtract mass times acceleration, you get zero.

**GEOFF:** But that's a constraint. Okay. So, if you have two sources of information, one of which is force and the other which is mass times acceleration, you'd like to have hidden units that see both those inputs and that say zero, no activity. And then when they see things that don't fit the physics, they'll have high activity.

**GEOFF:** That'll be the negative data. So that's called a constraint. And so, if you make your objective function be, have low activity for real things and high activity for things that aren't real, you'll find constraints in the data as opposed to features. So, features are things that have high variance and constraints are things that have low variance.

**GEOFF:** A feature is something that's got higher variance than it should, and constraint has lower variance than it should. Now, there's no reason why you shouldn't have two types of neurons. One's looking for features and one's looking for constraints. And we know with just linear models that a method like principal components analysis looks for the directions in the space that have the highest variance.

**GEOFF:** They're like features and it's very stable. There're other methods, like minor components analysis that look for directions in the space that have the lowest variance. They're looking for constraints. They're less numerically stable, but we know that it pays to have both. And so that, for example, is a direction that might make things work better.

**GEOFF:** But there's lots, there's about 20 things like that that I need to investigate. And my feeling is until I've got a good recipe for whether you should use features or constraints or both, what's the most effective way to generate negative data, and so on, it’s premature to investigate really big systems.

**CRAIG:** With regard to really big systems, one of the things you talk about is the need for a new kind of computer. And I've seen confusion about this too in the press. I've seen people talk about how you talk about getting rid of von Neumann architecture.

**GEOFF:** Yeah. You obviously want computers where the hardware and software is separate. And you want them to do things like keep track of your bank account. This is for things that where we want computers to be like people to process natural language. To process vision, all those things that some years ago Bill Gates said computers couldn't do, like they're blind and deaf.

**GEOFF:** They're not blind in deaf anymore. But for processing natural language or doing motor control or doing common sense reasoning, we probably want a different kind of computer. If we want to do it very low energy, we need to make much better use of all the properties of the hardware.

**CRAIG:** Your interest is understanding the brain.

**GEOFF:** Yes, but I have a side interest in getting low energy computation going.

**GEOFF:** And the point about the forward-forward is it works when you don't have a good model of the hardware. So, if, for example, I take a neural net and I insert a black box, so I have a layer that's just a black box, I have no idea how it works. It does stochastic things. I don’t know what's going on. The question is, can the whole system learn with that black box in there?

**GEOFF:** And it has absolutely no problem. It learns something different because the black box is changing what happens on the forward pass. But the point is it's changing it in exactly the same way as both forward passes, so it all cancels out.

**GEOFF:** Whereas in back propagation, you're completely sunk. If there's a black box, the best you can do is try and learn a differentiatable model of the black box. And that's not going to be very good if the black box is wandering in its behavior. So, the forward algorithm doesn't need to have a perfect model of the forward system.

**GEOFF:** It needs to have a good enough model of what one neuron is doing so that it can change the incoming weights to that neuron to make it more active or less active. But that's all it needs. It doesn't need to be able to invert the forward pass

**CRAIG:** And you're not talking about replacing back propagation, which has obviously had enormous success, if there is plenty of compute, plenty of power, then back prop is fine.

**CRAIG:** But, and this is speculative, I understand where you are in the research, but can you imagine if you had low power computer architecture that, that, could handle forward algorithms and, you scaled it?

**GEOFF:** I can imagine that it would be great. I've actually been talking to someone called Jack Kendall, who works for a company called Rain, who is very insightful about what you can do with analog hardware using properties of the circuits.

**GEOFF:** Using what for circuits, properties of the electrical circuits, natural properties of electrical circuits. Initially that was very interesting for doing a form of Boltzmann machine learning. But it's also going to be very interesting for the forward-forward algorithm. So, I can imagine it scaling up very well, but there's a lot of work to be done to make that happen.

**CRAIG:** Yeah. And if it did scale up very well to the degree that large language models have been successful, do you think that its abilities would eclipse those of models based on back propagation?

**GEOFF:** I'm not at all sure. I think they may not. So, I think that back propagation might be a better algorithm in the sense that for a given number of connections, you can get more knowledge into those connections using back propagation than you can with the forward algorithm. So, the networks with forward work better if they're somewhat bigger than the best size network for back propagation. It's not good at squeezing a lot of information into a few connections. Back propagation will squeeze lots of information into a few connections if you force it to. It’ll work. It is much more happy not having to do that. But it'll do it if you force it to, and the forward algorithm isn't good at that. So, if you take these large language models, so take something with a trillion connections which is about the largest language model, that kind of size, that's about a cubic centimeter of cortex.

**GEOFF:** And our cortex is like we’ve got a thousand times that much cortex. So, these large language models that actually know a lot more facts than you or I do because they've read everything on the web. Not everything, but an awful lot. The sense in which they know them is a bit dodgy, but if you had a sort of general knowledge quiz, I think GPT-3 even would beat me at a general knowledge quiz.

**GEOFF:** There's be all sorts of people it knows about and when they were born and what they did that I don't know about. And it all fits in a cubic centimeter of cortex if you measure by connections. So, it's got much more knowledge than me in, much less brain. So, I think back prop is much better at squeezing information, but that's not the brain's main problem for our brains.

**GEOFF:** We've got plenty of synapses. The question is how do you effectively get information into them? How do you make good use of experience?

**CRAIG:** David Chalmers talked about the possibility of consciousness, and you’re certainly interested in the possibility. If you understand how the brain works and you can replicate it, this kind of a model - let's imagine that it scales beautifully - do you see the potential for reasoning.

**GEOFF:** Oh, I see the potential for reasoning? Sure. But consciousness is a different kind of question. So, I think people - I'm amazed that anybody thinks they understand what they're talking about when they talk about consciousness.

**GEOFF:** They talk about it as if we can define it, and it's really a jumble of a whole bunch of different concepts. And they're all mixed together into this attempt to explain a really complicated mechanism in terms of an essence. So, we've seen that before.

**GEOFF:** Like a hundred years ago, if you asked philosophers what makes something alive, or even if you ask biologists what makes something alive, they’d say it has vital force. And if you say what is vital force? And could we make machines have vital force? They can't really define vital force other than saying it's what makes people alive.

**GEOFF:** And as soon as you start understanding biochemistry you give up on the notion of vital force. You understand about biochemical processes that are stable and things breaking down. And so, it is not that we cease to have vital force, we've got as much vital forces as we had before. It's just that it's not a useful concept because it’s an attempt to explain something complicated in terms of some simple essence.

**GEOFF:** So, another model like that is so sports cars have oomph, and some have a lot of oomph like an Aston Martin with big noisy exhaustion, lots of acceleration and bucket seats has lots of oomph. And oomph is an intuitive concept. You can ask doesn't Aston Martin have more oomph than my Toyota Corolla?

**GEOFF:** And it definitely has a lot more oomph. So, we really need to find out what oomph is. Because oomph is what it's all about, if you're interested in cars or fast cars anyway. But the concept of oomph, it's a perfectly good concept, but it doesn't really explain much. If you want to know, why is it that when I press the accelerator, it goes very fast, the concept of oomph isn't going to help you. You need to get into the mechanics of it how it actually works.

**CRAIG:** And that's a good analogy because what I was going to say is it doesn't really matter what consciousness is, it matters whether we as humans perceive something as having consciousness.

**GEOFF:** And I think there's a lot to be said for that. Yes.

**CRAIG:** Yeah. So, if this forward in a large model that scaled relatively low power consumption if it can reason…

**GEOFF:** There'll always be philosophers that say, yeah, but it's not conscious.

**CRAIG:** But it doesn't really matter if you can't tell the difference.

**GEOFF:** It matters to the philosophers. I think it would be nice to show them the way out of their trap they make for themselves, which is I think most people have a radical misunderstanding of how terms about perception and experience and sensation and feelings actually work, of how the language works.

**GEOFF:** If, for example, I say, I'm seeing a pink elephant, notice the words pink and elephant refer to things in the world. So, what's actually happening is I'd like to tell you what's going on inside my head. But telling you what the neurons are doing won't do you much good, particularly since all our brains are wire slightly differently.

**GEOFF:** It's just no use to you to tell you what the neurons are doing. But I can tell you that whatever it is my neurons are doing, it's the kind of thing that's normally caused by pink elephants being out there. If I was doing veridical perception, the cause of my brain state would be a pink elephant. I can tell you that.

**GEOFF:** And that doesn't mean a pink elephant exists in some spooky thing inside my head. Or it's just a mental thing. What it really tells you is I'm giving you a counterfactual. I'm saying the world doesn't really contain a pink elephant, but if it did contain a pink elephant that would explain my brain state, that plus normal perceptual causation would explain my brain state.

**GEOFF:** So, when I say I'm having the experience of a pink elephant, the word experience, many people think experience refers to some funny internal goings on. It's an experience. It's an, it's some internal. No. What I'm denoting when I use the word experience is that it's not real. I'm giving you a hypothetical statement, but if this hypothetical thing were out there in the world that would explain this brain state.

**GEOFF:** And so, I'm giving you insight into my brain state by talking about a hypothetical world. What's not real about experience is that it's a hypothetical I'm giving you, it's not that it lives in some other spooky world. And it's the same for feelings. If I say I feel like hitting you what I'm doing is I'm giving you a sense of what's going on in my head via what it would normally cause.

**GEOFF:** So, in perception, it's the world causing a perceptual state. With feelings, it's the internal state causing an action. And I'm giving you insight into my internal state by telling you what kind of action it would cause. Now I might feel like hitting you or anybody else or kicking the cat or whatever in which they say, instead of giving you any one of those actions, I just use a term like angry.

**GEOFF:** But really that's shorthand for all those angry actions. So, I'm giving you, I'm giving you a way of seeing what's going on in my head via describing actions I might do, but they're just hypothetical actions. And that's what the word feel means when I say I feel. Typically, if I say I feel and then say I feel like blah.

**GEOFF:** It's not that there's some special internal essence that's feeling, and computers don't have it. Computers are just transistors. They don't have feeling. You have to have a soul to have feelings or something. No, I'm describing my internal state via the actions it would cause if I were to disinhibit it.

**CRAIG:** From another human's point of view, if you were a machine and you were saying things like that, I would perceive it as you having feelings, right?

**GEOFF:** So, let's take the perception cases. It's slightly simpler, I think. Suppose we make a big complicated neural network that can do perception and can also produce language.

**GEOFF:** We have those now. And so, you can show them an image and they can give you a description of what's there. And suppose we now take one of those networks and we say, I want you to just imagine something and, okay, so imagine something. And then it tells you what it's imagining.

**GEOFF:** So, it says, I'm experiencing a pink elephant. That's experiencing the pink elephant just as much as a person is when they say they're experiencing a pink elephant. It's got an internal perceptual state that would normally be caused by a pink elephant, but in this case, it's not caused by a pink elephant.

**GEOFF:** And so, it uses the word experience to denote that. There you go. I think it's got just as much perceptual sensations as we have.

**CRAIG:** Although, the current state of large language models don't exhibit that kind of cohesive internal logic.

**GEOFF:** No, but they will. They will.

**CRAIG:** You think they will?

**GEOFF:** Oh, yeah. Yeah. I don't think, I don't think consciousness is - people treat it like it's like the sound barrier, that you're either below the speed of sound or you're above the speed of sound. You've either got a model that hasn't yet got consciousness, or you got there.

**GEOFF:** It's not like that at all.

**CRAIG:** I think a lot of people were impressed by you talking about using MATLAB

**GEOFF:** I'm not sure impressed is the right word.

**CRAIG:** They were interested, they were surprised. But what, is your day-to-day work like?

**CRAIG:** You have other responsibilities, but do you spend more time on conceptualizing and that could happen while taking a walk or taking a shower? Or do you spend more time on experimenting like on MATLAB, or do you spend more time on running large experiments?

**GEOFF:** Okay. It varies a lot over time. So, I'll often spend a long time, like when I wrote that paper about GLOM, I spent a long time thinking about how to organize a perceptual system that was more realistic and could deal with part-whole hierarchies without having to do dynamic setting up and connections.

**GEOFF:** And so, I spent many months just thinking about how to do that and writing a paper about that. I spent a lot of time trying to think about more biologically plausible learning algorithms and then programming little systems in MATLAB and discovering why they don't work. So, the point about most original ideas is they're wrong. And MATLAB's very convenient for quickly showing that they're wrong and for very small toy problems like recognizing handwritten digits. I'm very familiar with that task. I can very quickly test out an idea to see if it works. And I've got, I've probably got on my computer thousands of programs that didn't work well that I programmed in an afternoon and an afternoon was sufficient to decide that okay, that's not going to work.

**GEOFF:** Probably, that's probably not going to work. You never know for sure because there might be some little trick you didn't think of. And then there will be periods when I think I've got onto something that does work and I'll spend several weeks programming and running things and seeing if it works. I've been doing that recently with the forward-forward.

**GEOFF:** Let me say why I use MATLAB. I learned lots of languages when I was young. I learned POP2, which was an Edinburgh language. UCSD Pascal, a Lisp, Common Lisp, Scheme, all sorts of Lisps, and MATLAB, which is ugly in some ways, but if you're dealing with vectors and matrices, it's what you want. It makes it convenient.

**GEOFF:** And I became fluent in MATLAB, and I should have learned Python and I should have learned all sorts of other things. But when you're old, you're much slower learning language. And I'd learned plenty of them. And I figured since I'm fluent in MATLAB and I can test out little ideas in MATLAB and then other people can test out running on big systems, I would just stick with testing out things on MATLAB. There's a lot of things about it that really shape me, but it's also very convenient.

**CRAIG:** And you talk a lot about learning in toddlers and, is that knowledge base something you accumulated years ago or are you continuing to read and talk to people in different fields?

**GEOFF:** I talk to a lot of people, and I learn most things from talking to people. I'm not very good at reading. I read very slowly and when I come to equations, they slow me up a lot. So, I learned most of what I know from talking to people. And I'm lucky that I've got lots of good people to talk to.

**GEOFF:** Like I talk to Terry Sejnowski, he tells me about all sorts of neuroscience things. I talk to Josh Tenenbaum, and he tells me about all sorts of cognitive science things. I talk to James McFarland, and he tells me lots of cognitive science, psychology things. So, I get most of my knowledge just from talking to people.

**CRAIG:** At your talk at NeurIPS, you mentioned Yann - he corrected my pronunciation of his name - LeCun. Why did you reference him in that talk?

**GEOFF:** Oh, because for many years he was pushing convolutional neural networks. And the vision community said, okay, they're fine for little things like handwritten digits, but they'll never work for real images.

**GEOFF:** And there was a famous paper submitted to a conference where he and his coworkers, where he actually did better than any other system on a part particular benchmark. I think it was segmenting pedestrians, but I'm not quite sure. It was something like that. And the paper got rejected even though it had the best results.

**GEOFF:** And one of the referees said the reason they were rejecting the paper was because the system learned everything. So, it taught us nothing about vision. And this is a wonderful example of a paradigm, and the paradigm for computer vision was you study the task that has to be performed, the computation that has to be performed, you figure out an algorithm that'll do that computation, and then you figure out how to implement it efficiently.

**GEOFF:** And so, the knowledge is all explicit. The knowledge that it's using to do the vision is explicit. You had to sort it out mathematically and then implement it and sitting there in the program and they just assumed that's the way that computer vision is going to work. And because computer vision has to work that way, if someone comes along and just learns everything, they're not useful to you because they haven't said what the knowledge is, what is the heuristic you're using?

**GEOFF:** And so, it's okay, maybe it works, but that's just good luck. In the end we are bound to work better than that because we are using real knowledge and we understand what's going on. So, they completely failed to get the main message, which was that it learned everything. Not quite everything, because he didn’t write in convolution.

**GEOFF:** But the machine learning community, they respected him cause he's obviously a smart guy, but they thought he was on completely the wrong path, and they dismissed his work for years and years. And then when Fei-Fei Li and her collaborators produced the ImageNet competition, finally we had a big enough data set to show that neural networks would really work well.

**GEOFF:** And Yann actually tried to get several different students to make a serious attempt to do the ImageNet with convolutional nets, but he couldn't find a student who was interested in doing it. At the same time, Ilya became very interested in doing it. And I was interested in doing it. And Alex Krizhevsky was a superb programmer who put a lot of hard work with Ilya into making it work really well.

**GEOFF:** So, it was very unfortunate for Yann that it wasn't his group that finally convinced the computer vision community that, actually, this stuff worked much better than what you were doing.

**CRAIG:** You've now put this paper out there. Are you hoping to ignite sort of an army of people?

**GEOFF:** Yeah. I'm going to put some simple MATLAB code out there too.

**GEOFF:** Because there's a bunch of little things you have to do. Otherwise, it won't work. And the code needs to be out there. It's more picky than back prop. With back propagation, you just show people the equations and anybody can go and implement it and it doesn't need a lot of tricks for it to work quite well.

**GEOFF:** To work really well, it needs lots of tricks, but to work quite well, it's fine. With the forward-forward, you need a few tricks for it to work at all. The tricks are quite reasonable tricks, but once you put them in there, then it works. And I want to put that MATLAB code out there so other people can get it to work.

**GEOFF:** But I didn't want to put my very primitive MATLAB code out there because it's disgusting