**Geoff Hinton**

**CRAIG:** Hi, I'm Craig Smith, and this is Eye on AI.

This week I speak to [Geoff Hinton](https://en.wikipedia.org/wiki/Geoffrey_Hinton), who has lived at the outer reaches of machine learning research since an aborted [attempt at a carpentry career](https://nyti.ms/2tWNIfb) a half century ago. After that brief dogleg, he came back into line with his illustrious ancestors, [George Boole](https://en.wikipedia.org/wiki/George_Boole), the father of Boolean logic and [George Everest](https://en.wikipedia.org/wiki/George_Everest), British surveyor general of India and eponym of the world's tallest mountain. Geoff is one of the pioneers of deep learning and shared the 2018 [Turing award](https://en.wikipedia.org/wiki/Turing_Award) with colleagues, [Yoshua Bengio](https://en.wikipedia.org/wiki/Yoshua_Bengio), and [Yann LeCun](https://en.wikipedia.org/wiki/Yann_LeCun). A year earlier, he had introduced [capsule networks](https://en.wikipedia.org/wiki/Capsule_neural_network), an alternative to [convolutional neural networks](https://en.wikipedia.org/wiki/Convolutional_neural_network) that take into account the pose of objects in a 3D world, solving the problem in computer vision, in which elements of an object change their position when viewed from different angles.

He has been largely silent since then, and I'm delighted to have him on the podcast.

We began like so many of us do today trying to get the teleconferencing system to work. I hope you find the conversation as engrossing as I did.

**CRAIG:** I don't think I need to introduce you or that you need to introduce yourself. I do want to sort of recap what's gone on in the last year. It's been quite a year. Capsule networks had sort of faded from view, at least from the layman's point of view, and resurfaced at [NeurIPS](https://nips.cc/) last December with your introduction of [stacked capsule autoencoders](https://arxiv.org/abs/1906.06818).

Then, in February at the [AAAI](https://www.aaai.org/) conference, you talked about capsule networks as key to [unsupervised learning](https://en.wikipedia.org/wiki/Unsupervised_learning#:~:text=Unsupervised%20learning%20is%20a%20type,a%20minimum%20of%20human%20supervision.). And in April you revived the idea of [backpropagation](https://brilliant.org/wiki/backpropagation/#:~:text=Backpropagation%2C%20short%20for%20%22backward%20propagation,to%20the%20neural%20network's%20weights.) as a learning function in the brain with the introduction of [neural gradient representation by activity differences](https://medium.com/syncedreview/new-hinton-nature-paper-revisits-backpropagation-offers-insights-for-understanding-learning-in-477c54f2e7b6#:~:text=The%20researchers%20introduced%20neural%20gradient,states%20to%20drive%20synaptic%20changes.) or NGRADs,

**GEOFF:** I think it would be better if we started with capsules and we do three different topics.

We do capsules.

**CRAIG:** Okay.

**GEOFF:** Then we do [SimCLR](https://arxiv.org/abs/2002.05709). And then we do the NGRAD stuff.

**CRAIG:** Okay. Can you talk about your new capsule idea? Not new - a year old or longer now - but how that has influenced your research?

**GEOFF:** Okay. So, several things have changed and more things are changing right now.

So originally capsules, we used supervised learning and we thought it would be easy to get things working like that, even though I don't really believe in supervised learning. And last year we switched to unsupervised learning and we also switched to using [set transformers](https://arxiv.org/abs/1810.00825).

So what capsules are trying to do is recognize whole objects by recognizing their parts and the relationships between the parts.

So, if you see something that might be an eye, and you see something that might be a nose, the possible eye could say where the face should be and the possible nose could say where the face should be. And if they agree on where the face should be, then you say, 'Hey, they're in the right relation to make a face, so we'll instantiate a face. We'll activate the face capsule.'

So, there's various problems with that. One is the issue of whether you try and train it supervised or unsupervised, and it's going to be much better to use unsupervised because then you don't need labels. But the other problem, which we overcame with stacked capsule autoencoders, is that if you see say a circle in a line drawing, you don't know whether it's a left eye or a right eye or the front wheel of a car or the back wheel of a car.

And so, it has to vote for all sorts of objects it might be a part of, and so, you know, if it's the back wheel of a car, it knows roughly where the car should be and it can vote for, 'there should be a car there.' But of course, it might not be that. It might be a doorknob or it might be a left eye. And so, it makes lots and lots of votes.

And now what happens is every higher-level capsule gets a huge cloud of votes, nearly all of which are wrong. But one way to try and rectify that is to say, well, if any other capsule likes the vote, if any other capsule can make use of that vote, to make be part of this object, then route the vote there and don't route it to me.

And so that was the idea of dynamic routing; that you try and get all the bad votes to go to the places where they're good votes.

That's complicated to make it work. The alternative which we use in stacked capsule autoencoders s is to say, if you discover parts, suppose you discovered a circle and a triangle and a rectangle, you don't really know what they're parts of. There's many, many things they could be parts of. So, what you want to do is have them interact with each other a bit, and use the spatial relations between them to allow each part to become more confident by what kind of part it is. So, if you're a circle and there's a triangle in the right relative position to be a nose, if you're a left eye, then you get more confident than you’re a left eye. And that's what transformers are very good at.

Transformers have a representation of, in the case of language, a word fragment. So, it might be the fragment 'may,' which happens to be a whole word. And they don't know whether that's a modal, like would and should, or whether it's a month, like June and July. And so, what they do is, the representation of that fragment interacts with the representations of other fragments. And if there's another fragment in the sentence, for example, June, then the representation for May gets more month-like. Whereas if there's another fragment that's, would or should, it gets more modal like. And after a few layers of that, the fragments have been disambiguated. That is, you know, much better what each fragment is meant to be.

So, in language, that means you have a contextually sensitive representation of the word which s disambiguated between different meanings. In vision, if you have something like a circle, you'd like to know whether that circle is an eye, or the wheel of a car. And you can do that without yet creating a face or a car, by this interaction between parts. And in stacked capsule autoencoders, that's what we do.

We take the first level of parts and they all interact with each other. So, they become more confident about what kind of a part they are. Well, once they are more confident about what kind of a part they are, then they vote for what whole they might be a part of. And that way they can have far more specific, confident votes.

So, they don't make lots of crazy votes. Once you become convinced that a circle is probably a left eye, it doesn't vote for being the back wheel of a car. That means you've got far fewer votes to deal with it and so it's far easier to find the clusters. We made it so instead of trying to learn by supervision, by giving it labels, it learned to create a whole that was good at reconstructing the parts.

And so that's unsupervised learning.

**CRAIG:** At some point you need to connect it to language.

**GEOFF:** Yeah. So, all the learning in stacked capsule autoencoders, almost all the learning, is unsupervised. That is, you have these parts, and you recognize these parts, which are sort of templates that occur a lot – it’s a little bit more complicated than that, but - and then you recognize wholes that are combinations of these parts.

And the objective function is to find wholes that are good at reconstructing the parts, in particular, find wholes so if I tell you the pose of the whole thing, you can tell me the pose of the part. Like if I tell you there's a small face at 45 degrees in the bottom right-hand corner of the image, you can tell me that there should be a nose of 45 degrees that's even smaller in the bottom right hand corner of the image. So, the whole can predict the parts. And I didn't need any labels for that.

**CRAIG:** Right.

**GEOFF:** Now once you've done that, once you've got these wholes, you can then learn what they're called. So then supervised learning consists of taking wholes and learning their names. But you're not learning to recognize them when you're doing that, you're just learning what things you can already recognize are called, much like a little child learns to recognize cows and sheep. It doesn't know that cows are called cows and sheep are called sheep, and that's why it needs its mother to tell it. But its mother is not the one who tells it how to tell the difference between a cow and a sheep.

**CRAIG:** Yeah. Would this kind of unsupervised learning, in a larger system, also be able to make assumptions or inferences about relationships between objects or, or the laws of physics, for example?

**GEOFF:** Those are two somewhat different questions.

**CRAIG:** Yeah.

**GEOFF:** In the long run, we'd like it to do that, but let's return to the laws of physics later on when we talk about SimCLR.

Okay. for now, he recognizes objects by seeing parts in the correct relationships. And you recognize scenes by seeing objects in the correct relationships. In a scene the relationships between objects are typically somewhat looser, but yes, it can do that. It can recognize that objects are related in the right way to make a particular kind of a scene.

**CRAIG:** SimCLR [Simple framework for Contrastive Learning of visual Representations] came up later in the year. Can you talk about SimCLR and how that relates?

**GEOFF:** So that's a different learning algorithm. That's different in many ways. It's not, for example, focusing on the problem of dealing with viewpoint equivariance, that is, as the viewpoint changes, you get a representation that changes so that you can cope with viewpoint easily.

That's not the primary goal of SimCLR. What SimCLR is doing is saying, 'I want to learn to represent a patch of an image in such a way that other patches of the same image have similar representations.' So, what you do is you take a crop of an image and then you take another crop of the same image.

And you say, we're going to have a neural net that converts those crops into a vector representation, pattern of neural activities. And we want those patterns to be similar if the crops came from the same image and different, if they came from different images. If you just say, make them similar, that's easy.

You just make all of the vectors be identical. The trick is you have to make them similar if they came from the same image and different if they came from different images. And so that's called [contrastive learning](https://analyticsindiamag.com/contrastive-learning-self-supervised-ml/#:~:text=Contrastive%20learning%20is%20an%20approach,between%20similar%20and%20dissimilar%20images.). And [Ting Chen](https://scholar.google.com/citations?user=KoXUMbsAAAAJ&hl=en) in the Google lab in Toronto, with some help from others of us, made that work extremely well.

He wasn't the originator of the idea. In fact, the first comes from work I did with [Sue Becker](https://scholar.google.com/citations?user=WWVIu0EAAAAJ&hl=en) in 1993 or 92, and then later work I did in 2002. But we never really made it work well for images and other people revived the idea in 2018 and got contrastive learning working for crops of images. And then Ting Chen made it work considerably better, and that made people sit up. And so, what happens is once you've got this representation of a patch of an image, or, this neural net that can convert a patch from an image into a representation, such that you get similar representations with two patches coming from the same image, then you can use those representations to try and recognize what the objects are in the image. And that stage is supervised learning, but that doesn't require a deep net.

So, the idea is you do unsupervised learning by using this deep net to try and get the same representation or very similar representations for two different patches of the same image. And different representations for patches of different images. After you've used the deep net to do that, so Ting uses a [ResNet](https://en.wikipedia.org/wiki/Residual_neural_network), which is a [inaudible] kind of deep net, after you've done that, you then just directly learn to turn those representations with no extra hidden layers into class labels. So that's called a linear classifier.

It doesn't have hidden layers in it. And he does remarkably well. So, a linear classifier based on those representations that we've got by pure unsupervised learning with no knowledge of the labels can do as well now on [ImageNet](http://image-net.org/) as a supervised method, provided for the unsupervised learning, we use a bigger ResNet.

If you use a standard sized ResNet on ImageNet, you get a certain error rate and we can get pretty much the same error rate by using a bigger ResNet training it entirely unsupervised with no knowledge of labels. And then on top of the representations we extract, training a linear classifier.

**CRAIG:** And in that training, in one of the things I read, you talked about using augmented data.

**GEOFF:** Yes. It's very important when you do this. You can think of the two different crops as different ways of getting representations of the same image, but it's perhaps the major thing you do, but you also have to do things like mess with the color balance. So, for example, if I give you two different crops from the same image you can often recognize that they're from the same image by looking at the relative distribution of red, green, and blue - the color histogram.

And we don't want it doing that. So, to stop it cheating like that, you take two different crops of the same image and on one of the crops you change the color balance. And now it can't recognize that they're the same just by using the color distribution. And those are the two most important ones, doing different crops and changing the color balance.

**CRAIG:** Yeah. Is that augmentation something that the data scientist does in the data prep? It's not part of the model, the model doesn't automatically augment the data.

**GEOFF:** Well, it's not part of the data prep really, as you're training on the data, you'll, you'll get an image, you'll take two different crops of the image, and then you will augment those crops. You'll change the color balance.

**CRAIG:** Right.

**GEOFF:** So, you can't really think of it as modifying the data so much as given an image, you then get these crops with modified color balance, and you can modify all sorts of other things like orientation and stuff like that.

**CRAIG:** And that sounds very similar, from a layman's point of view, to what Yann LeCun is doing with video, where, where he, he, he takes a video and, and tries to predict what the next frame will be, in an unsupervised manner.

Am I wrong in that?

**GEOFF:** Well, it's not the same as trying to predict the next frame of a video. It Is the same, however, as trying to extract your representation from the next frame, that's easily predicted by the representation you extracted from the current frame. So that's contrastive learning. You can do contrastive learning for videos.

And you can say, you're really asking the question, ‘Did these two frames come from the same video?’ And that's a bit like asking, ‘did these two crops come from the same image,’ and you can use the same contrastive learning techniques for that.

**CRAIG:** Yeah. And then at AAAI, when you were talking on the stage with Yann and Yoshua Bengio, you talked about capsule networks as a form of unsupervised learning that has promise going forward. This SimCLR is another method. Are they related or can they be blended in making unsupervised methods more powerful?

**GEOFF:** They're somewhat different approaches at present. You could clearly try and combine them.

We're not doing that at present. Yeah.

**CRAIG:** In Nature. there was a paper, I believe it was in Nature, about sort of reviving the idea of back propagation as a function of learning in the brain. And you introduced this idea of, of NGRADs, neural gradient representation by activity differences. Can you talk about that?

**GEOFF:** Neuroscientists have been very skeptical about whether the brain can do anything like back propagation.

Well, one of the big problems has been, how does the brain communicate gradients? Because in back propagation, you need to change your weight in proportion to the gradient of the error with respect to that weight, whatever your error function is. And the idea is that you represent an error by the rate of change in neural activity.

And that's nice because it can have both signs, that is, neural activity can be going up or it can be going down, so you can represent both signs of error. And it also implies that the learning rule, which uses a gradient, is going to be something called [spike timing dependent plasticity](https://en.wikipedia.org/wiki/Spike-timing-dependent_plasticity#:~:text=Spike%2Dtiming%2Ddependent%20plasticity%20(,action%20potentials%20(or%20spikes).). That is when you change your synapse strength, you're going to change it in proportion to the error derivative.

And that means you're going to want to change it in proportion to the rate of change of the post synaptic activity. It is going to be the presynaptic activity times the rate of change of the post synaptic activity. And that's called spike timing dependent plasticity, which they found in the brain. And in fact, I've been suggesting for a long time that we use activity differences.

I had a paper with [James McClelland](https://en.wikipedia.org/wiki/James_McClelland_(psychologist)) in 1987, suggesting that temporal differences of activity be used as error derivatives. And that was actually before spike timing dependent plasticity had been discovered. By 2005, I got interested in activity differences again. And much more recently people have managed to make that work quite well.

I'm still somewhat skeptical. I think the brain could do back prop if it wanted to that way. It's a little clumsy and I'm now skeptical because I think back prop is too good an algorithm for the brain. So, the brain is actually dealing with a very different problem from what most neural nets are dealing with.

Most of the neural nets want to get a lot of knowledge represented in a modest number of parameters, like only a billion parameters, for example. For brain, that's a tiny number of parameters. That's the number of parameters you're having a cubic millimeter of brain, roughly. So, we have trillions and trillions of parameters.

But, we don't have many training examples. We only live for like a billion seconds or 2 billion seconds.

And so, we, we don't get much experience and we've got a huge number of parameters and neural nets mostly were in the other regime. They get lots of training and they don't have many parameters. Now, if you've got lots and lots of parameters and not much training data, what you want to do is somewhat different from backpropagation, I think.

So, I got very interested in the idea that there is one way of making this activity difference method work nicely; of trying to generate agreement between a top down representation and a bottom up representation. So, the idea is, you have, say, some hierarchy of parts. You look at an image, you instantiate parts at different levels.

And then from the high-level parts, you top down predict the low-level parts. And what you'd like to see is agreement between the top-down prediction, which depends on a larger context, and the bottom up extraction of a part, which depends on a smaller context. So, from some local region of the image you extract a part; from many of those parts, you predict a whole; from the whole, you now, top-down predict the individual parts. But those predictions of the parts have used more information because they're based on the whole, it got to see more.

And what you want is, agreement between the top-down prediction and the bottom up extraction of part representation. And, you want it to be significant agreement, so what you really want is on the same image, they agree, but on different images they disagree. So, if you take the parts from one image and the top-down predictions or another image, they should disagree.

And that's contrastive learning as in SimCLR. But it also suggests a learning algorithm for the brain that is somewhat different from back prop. And I got very excited. It's not quite as efficient as back prop, but it's much easier to put into a brain because you don't need to go backwards through many layers.

You just need to compare a top-down prediction with a bottom up prediction. I call it back relaxation. And, over many times steps, it will get information backwards, but it won't get information backwards in one trial. And back propagation sends information all the way backwards through a multi-layer net on a single presentation of an image and back relaxation just gets it back one layer each time, and it needs multiple presentations of the same image to get it back all the way.

So, I got really interested in back relaxation and whether that might explain how the brain was doing this learning of multi-layer nets. But then I discovered that sort of pure greedy bottom up learning did just about as well. I hadn't done the controls carefully enough. The bottom up algorithm that I introduced in 2006 actually worked as well as this back relaxation.

And that was a huge disappointment to me. I still want to go back and see if I can make back relaxation work better than greedy bottom up.

**CRAIG:** I see, and thus the [June tweet](https://twitter.com/geoffreyhinton/status/1273328639673806851?s=20).

**GEOFF:** Yeah, that's when I discovered that back relaxation doesn't work any better than greedy bottom up learning.

**CRAIG:** Is the assumption that, that the brain is so efficient that even if greedy bottom up can do it on its own that there wouldn't be this top-down function, or is it possible that that top down function exists as a, as a, an optimizer or something?

**GEOFF:** Well, you'd like this top-down prediction - and making it agree with the bottom up extraction - you'd like that to be better than just training a stack of autoencoders, one layer at a time. Otherwise it's not worth doing and training a stack of autoencoders, one hidden layer at a time, turns out to be pretty good.

And what's happened recently in these big neural nets is, deep learning really got going in about 2006 when we discovered that if you train stacks of autoencoders or restricted Boltzmann machines, one, one hidden layer at a time, and then you fine tune it, it works very well.

And that got neural nets going again. People then did things like speech. And vision on ImageNet, where, they said you don't need the pre-training. You don't need to train these stacks of autoencoders. You can just train the whole thing supervised.

And that was fine for a while. But then when they got even bigger data sets and even bigger networks, people have gone back to this unsupervised pre-training. So that's what [Bert](https://en.wikipedia.org/wiki/BERT_(language_model)) is doing. Bert is unsupervised pre-training. And [GPT-3](https://en.wikipedia.org/wiki/GPT-3) uses unsupervised pre-training. And that is important now. So, there was this on again, off again, where there was supervised learning and then I introduced unsupervised pre-training and then people said, 'Oh, but we don't need that. We just use supervised learning.' But now they're back to saying, 'Oh, but we do need some unsupervised learning.'

**CRAIG:** Right.

**GEOFF:** But the unsupervised learning algorithms are now getting more sophisticated.

**CRAIG:** Yeah. and again, the, the SimCLR is, at least as it relates to computer vision, is one method. The stacked capsule autoencoders is another method, and there may be others still. The learning and the brain, you know, I had a long conversation, about a year ago with [Rich Sutton](https://www.eye-on.ai/podcast-011) about [temporal difference learning](https://en.wikipedia.org/wiki/Temporal_difference_learning). And there is a view that, that that algorithm is, describes what's happening in lower brain function. and what you're talking about is cortex learning, and, it, at what point do they - are they completely different systems?

**GEOFF:** Yes. The big successes of computational neuroscience have been taking the work that Rich Sutton and others did on temporal differences and relating it to experimental studies on the brain and dopamine. [Peter Dayan](https://en.wikipedia.org/wiki/Peter_Dayan) in particular, was very important in showing the relationship between this theoretical learning algorithm and what's actually going on in the brain. But that's for reinforcement learning.

And I think reinforcement learning is kind of the icing on the cake. Most of the learning is going to be unsupervised learning. You have to learn how the world works and you don't want to learn how the world works by using reinforcement signals. You don't want to learn to do vision by stubbing your toe all the time. You want to learn to do vision some other way.

**CRAIG:** Yeah. This is giving you further insight into, into learning in the brain. I remember that that was really your initial impulse in getting involved in all this study.

**GEOFF:** Yeah. My main goal in life has been to understand how the brain works, and all of this technology that's come out of attempts to understand how the brain works, aren't really how the brain works. It's useful spinoff. But it's not what I was really after.

**CRAIG:** Is that all part of one general stream that you're pursuing that's headed to a particular goal?

**GEOFF:**  It's like this. if your research has been around for a while, you have a number of, kind of, deep intuitions about how things should be, and then you have particular projects that are like particular instances that combine those intuitions. And often projects that seem quite separate, eventually merge. But for now, the work on capsules is somewhat different.

Although all three of them could merge together. That is, if we, if we can get the idea of top-down predictions and bottom up predictions agreeing in a contrastive sense, that is, they agree well for the same image and they're very different for different images, that will fit in with stacked capsule autoencoders.

But it will also, it's also an example of contrastive learning as in SimCLR. It may also explain how the brain can learn multi-layer nets. So obviously I would like to, I'd like to have one solution to everything. That's what everybody always wants. it's just, you have to be more realistic and get parts of this. You can't get the whole thing all at once.

**CRAIG:** Yeah. With the rise of transformers in models like GPT-3 and now, in capsule networks, which is primarily computer vision, there's kind of a convergence between computer vision and natural language processing. How do you see that convergence progressing? And, and those are the two principle components of consciousness, if I'm not wrong. So are we working towards, a model that can perceive the world, an AI model that can perceive the world that's closer to a human perception in that it blends...

**GEOFF:** One of the big motivations of capsule s was that it would be, it would have representations more like the representations we use.

So, a classic example is, if you see a square rotated through 45 degrees, you have two completely different ways of perceiving that one is as a tilted square and the other is there's an upright diamond. And what you know about it is totally different, depending on which representation you use. Now, convolutional nets don't have two different representations of that.

They just have one representation of that. To get two different representations, you need something that imposes a frame of reference. And a very strong feature of our perception is that we impose frames of reference on things and understand them relative to those imposed frames. And if you get someone to impose a different frame, they'll understand things quite differently.

That was one of the big motivations for capsules. It's also one for computer graphics. So, in computer graphics, you represent a house with a particular coordinate frame. And then relative to that coordinate frame, you know, where the windows and the door are.

Again, that's the kind of representation we need to get into neural nets, if neural nets are going to get more like us at representing objects. At present deep neural nets are very good at doing classification where they do it a completely different way from people. So, they're relying much more on things like texture.

And they can see all sorts of complex textures that we aren't sensitive to. And that's why you get these adversarial examples where two things look totally different to us, but very similar to a neural net and vice versa.

**CRAIG:** Google has just filed for a patent on capsule networks. Is that, because of the successes of stacked capsule autoencoders?

**GEOFF:** You know, I don't know all the motivations for filing the patent, but I think the main motivation, which is true for most of the patent filings Google does, is protective in the sense they don't want other people to sue them for using stuff they developed. And so, Google really, isn't interested in making its money out of patents. It's interesting in make is making its money out of having great products.

And it doesn't want to be prevented from using its own research and its great products. And the patent laws have changed in such a way that the first to file - it's not the first to invent, it's the first person to file it. And so, you have to file patents, just protectively.

**CRAIG:** There's this paper, that that's, under review right now, [transformers for image recognition at scale](https://arxiv.org/abs/2010.11929).

Does that relate at all to this use of transformers in capsule networks?

**GEOFF:** Yes, it does a bit. So, what it is showing is that the kind of interactions between parts that work so well in things like Bert, for words, where you're getting word fragments to interact, also works when you're getting representations of patches of images to interact.

And it's also what's happening in stacked capsule autoencoders, where we have a set transformer that's getting the representations of parts to interact with one another and become refined. But then in stacked capsule autoencoders, we then jump to high level representations and we're doing it all unsupervised.

Whereas in the paper with the 16 by 16 patches, they're training it supervised to perform classification. They're not training it unsupervised. So somewhat different. But this general trend of, extract some pieces and then get them to interact so you get clearer about what the pieces are, which is what transformers do, that seems to be a very good way to go about doing, building layers of representation.

**CRAIG:** Yeah. I'm going to ask a question that I'll probably cut out because it's going to sound. Ignorant, but in, in transformers, both, in capsule networks, or in natural language processing models like Bert or GPT-3.

It relies on, massive, parameters, right? Billions of parameters.

**GEOFF:** Actually, less parameters than convolutional neural nets, but...

**CRAIG:** Okay. but, but it, it goes out into the world of the internet in this case and, and ingests all of this. And then it looks to me like it's a kind of search it's going out and, and, and finding, something that matches a representation.

And then fills it in with, with what's already out there. Is that wrong?

**GEOFF:** That's wrong. Yeah. You can go and find the closest thing. I mean, if you give it a story to complete, you can find the closest match on the web. And it'll do completions that are nothing like the closest match on the web. Basically, it's taking all this information in this data it's observed and it's boiling it down into these parameters that allow it to produce similar stuff, but not by matching to particular instances it's seen already.

**CRAIG:** And in the same way, capsule networks are creating a new representation

**GEOFF:** Yeah, and capsules networks should be able to deal with a new view of the same object.

**CRAIG:** Yeah. So where is your research going now? I mean, on these three streams or are there other streams?

**GEOFF:** My main interest always been unsupervised learning because I think that's what most human learning is. I'm interested in developing capsules further and in things like SimCLR,

I'm also interested in making distillation work better. So, the idea of distillation is you have a great big model and you've trained it on data and it's extracted the regular patterns in the data and got them into its parameters.

And now you want to train a much smaller model that will be as good as the big model, almost as good as a big model, but you couldn't have trained directly on the data. And so, we see this all over the place. So, insects are like this. The way insects, roughly speaking, the way most insects work is they have one stage that's just about extracting nutrients from the environment and that's called the larva.

Okay. And it's just an eating machine. And this great fat, ugly grub, or a caterpillar for a butterfly, just gets fat. That's its role in life. And then it basically gets turned into a soup. And out of that soup, you build the adult, which may look nothing like the larva. I mean, a caterpillar and a butterfly are very different things.

And they're optimized for different things. So, the larva is optimized for sucking nutrients out of the environment. And then the butterfly is optimized for traveling around and mating. And those are very different activities from sucking nutrients out of the environment. Now butterflies also get nutrients out of the environment, but they're not machines for doing that like larva. You also see it in mining - a nice Canadian example. So, if you want gold, first you take a chunk of the earth, then you convert it to pay dirt. and you have one way of doing that. And then you take the pay dirt and you heat it up very hot to try and get the gold out. I think that's how it works.

And the same for data mining. So, you've got a big set of data. And you you'd like to end up with a small agile model that can look at a new example and tell you what class it is, for example, but the kinds of models that are good at sucking structure out of the data are not necessarily the same as the models that are going to be small and agile and easy to use on your cell phone for making the right decisions.

And so, the idea is you use one kind of model for sucking structure to the data, a great big model. Once you suck the structure of the data, you get the great big model to train a small model. And it turns out the big model is much better at training a small model than just the raw data. And it's like an apprenticeship. Or like the way science works. Once scientists have done their research and figured out how things work, they can then teach school kids how things work. So, sort of any smart school kid can learn Newton's mechanics. But not any smart schoolkid could have invented Newton's mechanics. Once Newton invented it, which is kind of tricky, you can then explain it quite well. And you can, you can instill a model of it in a school kid. And so, the idea of distillation is we use great big neural networks for getting structure out of the data and then much smaller, more agile networks for actually using what we discovered. And it's now being quite widely used.

It's used in Bert, for example, to get more agile networks. But I think there's probably ways of making it much better. And that's another thing I'm working on.

**CRAIG:** Yeah. on, capsule networks in the, in the, AAAI talk. You all agreed, you and Yann LeCun and Yoshua Bengio, on the, on the, on the direction of your research in unsupervised learning and a lot of what Yann LeCun has been doing with video sounds similar to what you talk about both with capsule networks, and SimCLR.

**GEOFF:** Yann and I share a lot of intuitions. We, we, we worked together for a while. We have a very similar view of the world.

**CRAIG:** So, can you talk about how, again, at some point all of these ideas will converge? How his research relates to your research, particularly his research on video?

**GEOFF:** Our goals are very similar and the methods are quite similar.

And as we start applying SimCLR-like methods to video, they're going to get even more similar. So, this idea of contrastive representation learning seems to be very powerful and Yann's exploiting it. Ting Chen made it work really well for static images. And we're now trying to extend that to video, but we're trying to extend it using attention, which is going to be very important for video because you can't possibly process everything in a video at a high resolution.

**CRAIG:** Yeah. Yeah. And, and that's interesting, when, when you relate, this, machine learning to learning in the brain and certainly attention is, is, critical. Yeah. and can you talk a little bit about how, these models, even if they're not using the algorithms that you think are operating in the brain, how they are analogous to, to human learning. I mean, there's this huge amount of unsupervised learning that goes on with, as you were saying with capsules that at the end, there's a little bit of supervised learning that that puts labels to representation.

**GEOFF:** Let me just clarify that the first few versions of capsules we did were all using supervised learning because we thought that would make life easier, even though that's not really what we believed in, but now we're doing unsupervised learning and it works better.

And it's, it's ideologically far more satisfactory.

**CRAIG:** Yeah. but in, in the unsupervised capsule networks, at the, at the end, you connected it to language through ...

**GEOFF:** That's just to show that it's learned something sensible. I mean, obviously you want to connect to language. There's very nice work going on at Google now in robotics, where they're using deep learning for getting robot arms to do things, to manipulate things. But they're also interfacing it with language. So, you can tell a robot what to do, and the robot can also tell you what it's doing. And that seems very important. It also seems that if the robot can tell you what it's doing, like, you know, it's opening the drawer.

The objections of people like [Gary Marcus](https://en.wikipedia.org/wiki/Gary_Marcus) have to natural language processing, saying it doesn't really understand what's going on. But you know, if it says I'm opening the drawer and I'm taking out a block and he opens his drawer and takes out a block, it's very hard to say it doesn't understand what's going on.

**CRAIG:** we mentioned at the beginning the laws of physics, learning the laws of physics, and you don't need language to learn the laws of physics. You do need a linguistic interface, to, to look at a tree and a car and be able to identify them as a tree in a car.

Can you talk about learning, something like the laws of physics that doesn't require language to be attached to it, but nonetheless, there's learning that takes place?

**GEOFF:** Yeah. At high school you may learn the laws of physics. We learn sort of common-sense physics. So, we learn, you know, if you throw something up it comes down again and if we're good, we learn how to throw a basketball so it goes through the hoop. And that's a very impressive skill cause you're throwing it from like 20 feet away and you have to get it right to a few inches. That's an amazing thing to be able to do.

And we, we don't learn that by being told how to do it. We don't run that using language at all. We learn it from people who say trial and error, but we're understanding how the world works just by observing the world. also, by acting in the world. So just passively observing the world will allow you to understand it, but it's not nearly as good as acting in the world.

And in fact, if you think about perception for robots or wander around an act in the world, it changes your view of high perception should work. So, if you're just taking images or videos and just passively processing them, it doesn't make you think about attention. But as soon as you have a robot, that's moving around in the world, it's got to decide what to look at.

And the sort of primary question in vision is where should I look next? And that's been sort of widely ignored by people that just process static images. Attention is crucial when it's sort of central to how human vision works.

**CRAIG:** Can you sum up a little bit, everyone likes to hear about convergence of all of these things, you know, convergence of, of computer vision with, natural language processing, convergence of unsupervised learning wit. supervised learning and reinforcement learning. is, is that beyond what, what you're really focused on? because you're focused on, on, the basic research, not necessarily, building models that pulls it all together.

**GEOFF:** Let me just say something about supervised learning versus unsupervised learning, because, it sounds like a very simple distinction. but actually, it's very confusing. So, if you ask. When, when a kid's mother says that's a cow, we tend to think of it in machine learning as the mother supplied a label.

But what's really happening is this. The child has some sensory input and the child is getting a correlation between the visual sensory input and the auditory sensory input. Now the top level, the auditory thing gives you the word cow and the visual thing gives you whatever your visual is and you learn they go together. But actually, supervision when you actually get it in reality, it's just another correlation and there's, so it's all about complex correlations in the sensory input, called supervised and unsupervised learning.

And then there's correlations with payoffs. And that's reinforcement learning, but I think the correlations with payoffs don't have enough structure in them for you to do most of the learning. So most of the learning's unsupervised.

**CRAIG:** Okay, well, let's, let's leave it there. I really appreciate it. And, it's been a fascinating conversation and I'll edit it down to be a coherent on my side.

**GEOFF:** Bye for now.

**CRAIG:** Yeah, bye-bye. That's it for this week's podcast. I want to thank Geoff for his time. If you want to learn more about the episode today, you can find a scrolling transcript on our website, www.eye-on.ai.

And remember the singularity may not be near, but AI is about to change your world. So, pay attention.