CISE Distinguished Lecture Series

Please stand by for realtime captions.
Let's start recording the meeting. Welcome to the National Science Foundation distinguished lecture series of CISE. We're delighted to have Geoffrey Hinton here, who, won the train award in 2018 for his work on learning and artificial neural networks. The citation is for conceptionual and engineering breakthroughs that have made deep neural networks a critical component of learning. I think a critical component of computing. Beyond this making him a critical component of computing, essentially establishing the whole intellectual foundation of the field and everything it tells us about learning in general. He is an engineering funnel at Google, as well as the Vice President and Chief Scientific Adviser on The Vector Institute at the University of Toronto and holds a doctoral degree on artificial intelligence from the University of Edinboro. His list of awards is pretty staggering. I thought two of the most interesting were companion of the order of Canada. Canada's highest honor and fellow of the royal society in the U.K. The society that goes back to the days of Newton. He needs little introduction as basically the person, I think, responsible for, I said, shaping the intellectual foundations of the artificial neural network machine learning and keeping it alive through many years, where it was a not-so-popular couple of interesting notes I saw in his biography. He's the great, great grandson of George Buhl, the founder of Symbolic Logic, and his middle name Everest honors his relative, Colonel Sir George Everest after whom the mountain is named. He will take us to new intellectual peaks. Before he starts out, remind people that at the end of the talk, we'll have a moderated question session. You can type your questions into the Q&A window. Tonight type them into the chat window but the Q&A window. I expect a flood of questions. If you see questions that others have asked that you would like to have chosen to be asked, a click on the thumbs up. That will help us sort through them. With that, Dr. Hinton, please proceed.
Okay. Thank you for inviting me to give this talk. Was once funded by you and I owe you guys. I need to apologize. A lot of people will know the technology of neural nets and others know less. I am going to go too slow for some people and too fast for other people, which on average, should be about right.I can't advance my talk anymore. There. I am going to to be talking about imaginary system called GLOM. It doesn't exist. Gloom is just a -- GLOM is my way of talking in the context of a system. This system is going to try and combine a lot of recent ideas in neural networks and combine ideas from transformers, which have had the most success of modeling natural language in something called BERT, WHICH IS MORE SUCCESSFUL IN VISION AND WILL HAVE IDEAS OFUP VISUALIZED LEARNING OF VISUAL REPRESENTATIONS. GO BACK TO 1992 and started working very well quite recently. And it's going to learn ideas about using implicit functions in graphics I will briefly explain each of those advances and then combine them into a system crawled glom. If wait works, it will be a perception that is more human-like. I want to start by spending time convincing you about something, about human perception. People tend to think that Cartesian frames, a mathmal catup ring is, that is quite useful for doing geometry with is not how people work. I actually believe it's how people work. That we impose rectangular coordinate things on things and our understanding of objects is relative to the rectangular coordinate frames we impose. In order to convince you of, that I am going to do a demo. An old demo and to get the full benefit, you have to do the task yourself. I will tell you what to do and you actually have to do it. Since zoom, nobody can see you getting it wrong. He's the task. Imagine a wire-frame cube resting on a table top in front of you. Imagine a foot along one side and made of Mattt black wires. -- Matte black wires. You can do that, right? Imagine the body diagonal going from the front bottom right corner to the top corner through the center of the cube. This is the bit you have to do. You put one hand in front of you, one finger of your left hand that represents the front bottom right corner. And there is another finger of the other hand that represents, here we go, that represents the top back left corner. Now what you do is you imagine rotating the cube so this finger is vertically above that finger. Okay. Now you have the, what was the top back left and it's vertically above the front bottom right. Okay. Taking away the hand that is the bottom corner that is resting on the table top still, you have to point with the finger, the other finger, you have to point to where the other corners of the cube are. Put your fingertips in space to where the other corner of cubes are. You have to do that yourself. I am going to force you to point out where you think the other corners are. And then I will tell you what most people do. The response is to say, well, here, here, here and here. The thing is that people point out the four other corners and then you are to recount the corners of a cube in the normal orientation and you will discover there are eight corners. What people pointed out was four corners. So, that is going on. The fact is that I forced you to use a different coordinate system for the cube by taking that axis and making it vertical. I said, relative to that corner system, where are the corners of the cube? The other corners? This is defined by two corners. People simply can't do it. Chris Ludquist can do it and a few people do it, but most people say, here, here, here and here. They point out four or point out the not part of the same height, the four, but they

normally point out four corners. So, one explanation is that when you rotate the cube mentally, the center is so great, two of the corners fly off and you're unconscious. That is not my theory. My theory is that you're using rectangular coordinate frames and I forced you to use a unusual frame and you don't know where things are in that frame. What is interesting is what people point out. They point out the four corners typically at the same height along this access. And so they, what they pointed out is two square based pyramids stuck base-to-base. My top fingers on the corner make one period and the other fingers make another pyramid and two periods stuck base-to-base. That is called an optahedron. It's not a cube, but it's got an interesting property, the same simtris as a cube in the normal orientation. And that is if you substitute the corners for faces and faces for corners, a cube turns into an optahedron and you managed to preserve the full four rotational symmetry. What you really know about cubes which, is obviously the standard orientation, you have this four-field symmetry or pair of symmetries, and that is what I give you. What you preserve is that. I will show you a picture where they are. The colored Rods of the edges of the cube that don't connect to your two fingertips, and those the six corners and they form a zig-zag ring. I have here with me, if I back up from the camera, I am not sure -- yeah, here we go and this is the cube. And getting a rotation for you to see. Hopefully, you can see that. You see that zig zag ring. I have a picture of it here. I want to use it to show you one more thing about human perception. Not only do we use rectangular coordinate frames, but we pause things. So if you look at the six Rods, I have colored them in such a way that you naturally pass a green flap, a blue flap, and a red flap. I will call the pause of the six rings a crown. It's three flaps that slope upwards and outwards. Okay. That is one way of seeing the six Rods. There is a different way of seeing the same sex roads -- the same six Rods. You can see the green flap is the same as before and now you have the ends of a blue rectangle and a red flap that slopes upwards and upwards. Seems like a bit of a puzzle. When you see it like this, you have a completely different internal representation. If you see it like this and I wish to take the image away, you would still know there were two parallel lines there, when you see it as a green flap sloping up, the red one down and a central rectangle, you're aware lines a&d are parallel but not other parallel lines and that is because A&d line up. If you look at E&B, THEY'RE ALSO PARALLEL. YOU'RE NOT aware of that when you pass it here. That is because they're in different parts, and because they don't line up with the natural coordinate frames they're in. One other thing about this cube and octahedron. Sorry, sexahedron, it's a cube but seen as a different way. As a sexahedron, you're not aware of the right angles. If I deform it, you don't notice when it's a right angle or not. When I see it as a cube and deform it by 2 degrees, you are very la cutely aware it's a deformed cube. In other words, you only notice the things at right angles when they line up with your rectangular coordinate system. If I get you to impose a rectangular system and the edges don't line for it, like for the flap E-F, the green flap, you're not aware if the angles are acute. The vertex of the E-F flap is a right angle. You're not aware of that because you're seeing it as a triangle. Enough said. The upshot of that is a fun demo. It shows that people haven't got a clue where the pieces are if they use a different coordinate frame and they use rectangular coordinate frames and that, pausing images and they can pause them differently and then they're aware of different things. If we take a standard neural net, a convoolutional neural net to recognize the objects, none of those are true. It does use intrinsic coordinate flames, not -- frames not explicitly and doesn't pause things, at least not explicitly, and it doesn't have two completely different ways of seeing exactly the same image. Notice this is not like a cube where the two different interpretations correspond to different 3D structures. The different pauses of the same structure with the same truth conditions, but they're different senses. Fully on sofical terms. -- philosophical terms, different ways of seeing the same thing. Convolutional nets don't have that. If you want neural nets to work like people, they better have that. I just said that. Okay. Here's a good old-fashioned structural description. I have nodes and connecting the nodes. This is for the crown interpretation of the six Rods. And the edges, what I put on the edges are the coordinate transformation between the coordinate system you impose on the whole thing like the crown and the coordinate system you impose on one of the parts like the green flap. If you think about the relation between the two things, there is a transformation of a coordinate system that is RWX. And that handle doesn't change as I change viewpoints. If I, if I bend the Rods, that will change. The relation between the crown and flap will change. But if I change my viewpoint, all of those things written on the arc are stable across the viewpoint. And that is why they're very good ways of describing a shape. They completely viewpoint independent. Here's the alternative pause I showed you. And this is something else I call a mental image. It's the underlying structural description, but associated with that structural description in those blue boxes, we've got the viewing transformed. That is for each of the nodes, we know the relationship of the intrinsic frame of reference, for example, the crown, to the camera or the -- and that is the relationship to the viewer. What you will notice is that if I tell you what is in the top blue box, the RWV, then if you multiply that by RWX, you get RXV. You get the relationship of the flap to the viewer. In other words, if you have a structural description like this and I give you the viewpoint on one of the nodes, you can propagate that viewpoint information to all of the other nodes. That is a very useful thing to do. That is what

I think a mental image is like. And it's useful because if I was to ask you what is the relationship if you see yet this way, between edge A and edge D well, in different parts they are in and this is a different part in this graph. It has four matrixes in this. If you can propagate viewer information to see it mentally and you have a viewpoint and everything,en in to know how A relates to D you need to look at the relationship between RAV and RDV so you have it down to a simpler operation. That is the conditions under which you use mental imagery. Okay. What I said, so far, all fits in with good old-fashioned -- you have a nice structural description. We impose relations to the viewer to do competitions more easily. But it's not really challenging conventionally. And indeed this picture was made in 1979, the hey day of conventional [Inaudible] Now, what I want to talk now is about isn't something that has taken me 40 years to figure out, which is how do you get this kind of structural description into a neural network? The problem is that the neural networks don't work like conventional computers. They don't have random access memory. You can't create data structures and allocate the data structures. What a neuron does is tied up with the weights it has. The weights of a neuron are that's things you can quickly change. You can't create a node. Okay. But we do need to create a node. We have to be able to create the node in the parse tree, but it can't be by setting aside a piece of memorior a group of neurons to act as that node. In the computer, we set aside memory and give that to the memory of other nodes and we would be done. You can't do that with neurons. What do you Don stead? You have to deal with this problem that every image is going to have a different parse tree. You have to dynamical create a representation of the parse trees. You have to dynamical create something like a pointer. This is a new thing you have to represent. And other things have got a node connected to the same new thing.But you can't set aside the neurons to be a part of hit I am going to step back and describe the transformer to you. There is one special thing about transformers. In standard neural nets, the activity of a neuron depends on the scale product of the imperfector with the weights of that neuron. You're multiplying vector activities with vector weights and that is what determines whether a neuron gets active. In transformers and something called capsules, which I have been pushing, think of capsules as a transformer that doesn't work very well. The idea is that neurons will get active based on a product of, a scaled product of activity vectors and that gives you have did of rent activities. If you can take a scale of product activity vectors, you can determine whether they have the same value or not to solve the x or problem without any hidden units. That is the fundamental innovation of transformers, to use scaled products of activity vectors. And I would show you how transformers relate to convoolutional neural nets for processing sentences. Suppose we have a string of four words. Nowadays, you use word fragments but let's not have that complexity for now. In the neural network, the first thing you do with a word is convert it to a distributed representation called an imbedding. So vector activities. I will call that the wordvec. So four words come in and you covert them to four wordvecs. The idea is to have a number of layers. In each layer, we will try to refine the word vec. We're going to make the word, vec, that initially is not quite sure what the meaning of the word is. We're going to make it more and more confident about the interpretation of that particular word. Let's suppose the third word in this string of four words was the word may. Let's suppose we removed all capitals. Now, we don't know whether may is a month or a model or may be it's a woman's name. We just don't know which it is. Let's suppose the word next to it is Jew. Now it's less like well to be a model and might be a month. It might be a woman's name still, but it's likely to be a month. The idea of those four heavy, black arrows is that you look around at nearby words and they help you revise your word vector to make it less ambiguous and more appropriate to the context. So, if the third word came in was may, the initial word vector would be ambiguous between a modal and a month. After you have seen, I mean suppose the previous words are Friday the 13th off, then you would be pretty confident it was May that came next and you ambiguated it to make the word vector much more correct for a month than a modal. That is how convolution nets would do it. Notice with the convoolutional net, the nearby words are having an influence and they're having an influence that is independent contract. The word, the second word is going to have an influence on the reinterpretation of may, the refinement of the imbedding vector, but it's not going to depend on what the second word is. Transformers do a better job of this. In a transformer, what you're going to do is the word is going to come in. You will do something quite complicated. I will make use is of transformers later but a simpler form of them. The word "vector" is going to give rise to a key value. And in should -- a key query value. In order to rise the vector, there is a missing black arrow from the vertical black arrow from the word vector itself to the next time stamp. So the meaning depends on the input word. If you look at the three other black arrows, what is happening is that they can have different strengths, depending on the match between the query value of the third word and, for example, the key value, the key of the 60 word. If the query and the key match, then the value associated with the second word is going to strongly influence the new imbedding of the third word in the sentence. If they don't match, there is not much influence. The idea is that the third word, which is may, creates this query. The query likes to see what other keys it matches. If it matches the keys of the other word, the word is likely to influence a stronger vector, then now imbedding.

You make these to one by simply saying that you're going to take the scaler product of the query vector of the third word, for example, the key vector of the second word, and then you're going to exponentiate that. If there is a good match, it's going to get a bigger impact than a poor match. That is how transformers enable the context influence how you revies bedding -- imbedding of a word. But, there is only relevant things in the context. Any irrelevant things won't have much effect. That is the general idea of transformers. Now, another site I am going to talk about, contrastive learning of representing as without labels. The idea of contrastive self-supervised learning is that the visual system probably extracts a whole bunch of structure from the input data without requiring labels like that is a cat and that is a dog. We have few of those. One interesting objective function is to try and reconstruct the input image and develop things called auto encoders. Oop alternative objective function, it was slightly motivated by the fact that I was once a philosophy student and was forced to read Kent "and Kent went on about space and time. A different objective function was to say, why dent we make the input and try to extract properties that can adhere across space or time. So with sue Becker in 1992, I published a version of this. It didn't get discovered very well. Now, it works well. I will talk about one version of the app in Toronto. There are other versions. They're getting better all of the time. Every few weeks, someone makes it look better. This is now good enough to compete with supervised learning of labels. You do unsupervised learning for representing as. Once you develop the representations, you make a simple supervised system that uses the representations you already decided to and tries to classify things. Opposed to having to chow on the filters of the labels. It's making things coherent across space here. I will describe in more detail how one of the systems worked. This is developed in Toronto. The idea, you take an image "X" and you make two different transformations of the same image. These transformations, the ones that work best are, you take two different crops. You're looking at different patches, quite possibly overlapping from the same image. You also mess with the color bounds. And then, you take the two transformed versions of the same image. You put them through one of the deep neural nets, like a residual net developed for doing ultraclassification. We're not going to give that net any labels. That is called "F," and that is a big deep neural net. You will get a representing a for the first image and for the second image. Hi and H J. Then you're going to do something else. You will compress the representing as, project them down, which I am not going to dwell on here. And you will get two smaller representing as called ZI, ZJ and your objective for learning is to maximize the agreement between ZI AND ZJ. You want ZI to be similar to ZJ. You can do this. If you do it in the obvious way and say make the vectors similar, what is going to happen is everything is going to collapse. And there are going to be the four image. You really mean to make them agree when I show them two of the same image and not agree when I show them two batches of the different images. That is why the learner is contrastive. You try to make investigation zi similar to ZJ. If ZI and JZ are different, don't do anything. If they're similar, then make them more different. That is what stops it. And that works nicely. Develops nice representations of image patches. I will leave it at that. And these representing as are good because you can put a simple linear classifier on top and it does very well. New, contrastive learning has a problem, which is you want to use it for recognizing objects. Suppose one patch of the image contained opposites of class A and B. You don't want the sane representing a of the two patches. You want the first patch to know that part of it should be the same as the representation of the second patch, the part that contains object A and part of it should be different. It has to have more say about what it's similar to and not similar to. That is like the difference of the convo Lieutenant Governorral transformer I talked about. It's more like transformers for doing this contrastive learning. My system called gloom is designed to do that. -- GLOM is designed to do that. Glom a way of discovering spacial coherence that is smarter than stronger condraftive learning. A number of other groups realized recently that you need to be selective about what you contrast with. You need to have things have their own idea, what they ought to be similar to. In particular, a paper by a bunch of authors at Berkeley, one of who is Efross, uses a technique like this in a recent paper to track objects and images much more successfully than before. So, before I talk about the details of GLOM, I want to make disclaimers. Vision is a sampling process. Human vision, in which you have a sequence of intel gent fixation points that were chosen so you don't have to focus on everything high resolution. You're sampenting the optic array to normalize that vision. I am going to focus on what happen if you make one fixation and it's your first fixation and you have no prior expectations of what you will see and by making the first fixation, I get rid of prime knowledge of this particular scene, and I don't have to worry about the issue of how you decide where to fixate. Ignoring all that. So, I mentioned this briefly already. In sibombic AI, what you do to represent a note is you set aside hardware. Some ram, and give it pointers and you don't. I had a system called capsules where the idea was because you death penalty dynamical allocate neurons, you will alionicate them ahead of time and you will allocate a bunch of neurons for each possible node in the image, in any image so most of the groups and neurons won't be used most of the time. For each image, you activate a small subset of the captions and dynamical hook them together. To represent the past node. We goat them to work but it never worked great for bigger datasets. In flop, we will use a new Wau -- GLOM, we will use a new Wau to represent ions of agreement. We'll see what that means. There is a biological analogy here. Here's what we expect that is so familiar that it doesn't seem weird anymore. Every single cell in your body has the instructions for being anything. It has unlimited potential when it starts out. So your brain cells know how to be liver cells and that seems completely crazy. The actual environment it's in determines the proteins it builds, which Genes it expresses. Cells differ in their vector of protein expressions. They have the same knowledge. I want to do the same with capsules now. Old-fashioned capsules were about a particular type of part and had the knowledge for that particular type of part. The new capsules now have -- not specific to a -- each capsule has to know about everything, which is quite demanding, but biology is good at dealing with the massive replication. It does things on an atomic scale. Every capsule, when I know everything, will be a universal capsule, and this seems very wasteful, but we're going to see it's convenient and it's not really wasteful. You need the computation it provides you with. So, we're going to make image likes on bilike. Imagine an image has a few thousand locations. If you're a neuroscientist, this is like hypercomalls in the visual cor tech. The weights -- hypercortex. The weights are like DNA. In all of the columns, we will replicate the same knowledge. And that is what we do in convolution. We duplicate it in every like. The complete victors is like a vector approached in a cell. So a bunch of cells with a similar vector approaching from an organ. Right, that is what makes an organ an organ. The cells are expressing the same proteins as you will understand. Similarly, an object is going to be a bunch of locations that have similar activity vectors. Some will have similar activity vectors different from one another. Overall, the complete activity vector location is going to be similar for all parts of an object. That is the analogy there. This is the GLOM architecture. I am showing you three adjacent levels. This is for a single locationwe're going to apply it to static image and to a video that is very boring. Okay. It's designed to deal with video, not static images, your vig. In this architecture, frame one comes in. You get a low-level representation. Level l one representation. A time stamp later that will lead rise to a representation of level L and a time stamp giving representation of level L plus 1. The blue arrows are what I call the bottom-up network there are at least two hidden layers in each of the blue arrows. Getting from level one to another is complicated. It's going to do coordinate transformations and needs to be complicated. From level L to plus one is quite complicated. This is a recurrent net. If you look at the three diagonal blue arrows in the bottom row between level L and minus 1 and level Well, that is the same network being used again. The weights. Right? A vertical time slice. The time is time. There is also a top-down level. If you look at level L plus one, there is a red level going to level L, and that is happening all of the time, too. You're predicting a level from the level above. And this is a recurrent network that operates over time. A fixation comes in and for multiple time stamps, you're settling on an interpretation of that. The settling involves lower-level things activates high-level things and high-level things activating lower-level things. That is contextual influences.It's getting another thing that is not shown here, which is, this is all for a single location. So this is all going on within one hypercolumn and it's going to get contextual influences from other locations, where the transformer aspect comes in. Sorry. The four influences on the level are the bottom-up, top-down and within the same location. There is previous embedding but then there is an average of the embeddings at nearby locations. So that is contextual influence. That is going to be an attention-weighed average. You don't willy-nilly average with the nearby locations. You look to see if the nearby locations agree with you. If they agree with you, you average with them. This is kind of well-known in America, called an echo chamber. You only accept opinions from people who already agree with you and what happens is that you get an echo chamber where a bunch of people have the same opinion. And so, GLOM uses that in a constructive way. What I am showing you now are the virtual columns here. They correspond to other columns in vertical areas and those are six adjacent locations. Imagine six likes in a row. The first like, the leftmost column, the data comes in and you form the lowest level embedding. You may want to go through a convolutional match and other stuff before that embedding and that is the blackiro pointing down and to the right. That is meant to be a vector representation. It would be a activity in hundreds of units. Of the nearby like, things are different in the image. That is the lowest level. Everything is different. Now, if you look at the leftmost two columns here, you will see that the next level, they have the same representation. What happened is the system discovered a local part. The part spans the two locations. What it's doing is giving the same vector representation to both of the locations. As you go up one more level, you can see the first three locations. They're an hour apart, and they have the same vector representation at that level. Of course, they have different representing as at the lower level. If you -- representations at the lower level. If you look at the green arrows at the right, that is a different part. The red arrow discovered a nose and the green arrows as discovered a mouse. At the next level, all of the black arrow are the same. The nose has predicted a face and the mouth predicted a face and they predicted the same face. That is a face with the same pose and same relationship to the camera. At the next level up, they agree on what the face is you will notice if you can get this to work and form the echo chambers and they're bigger as you go up, then you do have a representation of a parse tree and can represent more than a parse tree. It has no problem with things like sentences like may be this will glow gloom Aryans down. It's -- glomarians down. You can do that. You can have discontinuous items. So, the way the parse tree is represented here is is that of the obvious level, you have a big Island. The parse of the object is smaller islands. The subparts are smaller islands and so on. I can take this bunch of vectors and find light to turn it into a tree and draw a black arrow at the top. A node for the green and red arrows and so on. That is what I would do to illustrate the parse tree it found. This representation in and of itself is perfectly fine. This has all of the information you need from parse tree. Every location knows which node it's a part of at each level. And parts of the same object can talk to each other because they're using the same vector. If you ask what happened to the address we would have used in symbolic AI to represent the node, well, look at that black vector at the top. That is the same for all of the parts of an object. That vector is like the address. It's the thing we create. It's a vector of activities. It's not an -- like in the computer. It's not symbolic in that sense. It's a vector of activities. You are able to judge whether that is the same as another vector, you can do that by seeing if they have a big scale of product. Also, it's able to point to things. So the one bit I haven't shone you is how you get one of those black vectors at the top to point to one of the red vectors at the level below and get the very same black vector to point to one of those green vectors in a different column. And this seems like a bit of a problem. You have the same representation of the object and it has to point to different parts in different places. The noise is not helping us. This is the deterministic thing we want. We want the black arrow to give us the red arrow in some locations and the green arrow in other locations and that is where implicit funks come in. -- functions come in. This is a recent example of an implicit function decoder. Imagine you have four pixels in a row at locations one, two, three and four. Imagine the four pixels have a greatest of intensities. The intensity of pixel one is fx 1, and different from the second one, FX 2 because this is a consist gradient, we think they're part of the same surface. We would like to give all four pixels the same representation. Despite giving them the same representing a, we would like to be able to reconstruct from the representation the activity of a pixel. Well, suppose the pixel is determined by the function F. The F of X equals X plus B. What you need to do is give each pixel a representation that is just the coe fishingience in the e -- coefficience of the equation. They all have the same representation vector. They don't all reconstruct the same thing. The decoder gets to see the pixel representation, AB, but it also gets to see the pixel location, X 1. The decoder puts together the AB with the X1 and the decoder has to implement the funk of IF, XA equals FXB. Things can have the same representation and give rise to different things in different places, provided the decoder is told about the place. So, let's go back. Let's suppose that that black arrow is a particular offense. There is a deafination of face and knows the pose of the face. The relation of the face to the camera and that relation and the intrinsic -- take the first column on the left where the black arrow gives rise to the red arrow. If that is a face, suppose the red arrow is the nose, right? And what you have to do is take the face vector. Take the image location of the column you're in that is where we are in the image. The face vector contains the relationship, the 3D relationship between the intrinsic coordinate From of the face and that the of the image. -- and that of the image. If you know the place in the image, let's say 3D image, not just space and images, you know how the face is related to the image. And you can take the like and figure out where that location is in the coordinate frame of the face. Okay. We know it's one of the nose locations. And that will be where you find noses and if you look at the right, the black vector is the same face vector and you're asking the location now, where is the other location to the face, the coordinate frame of the face to be a mouth? And now you will expect to and a mouth there. The top-down network can take the black arrow and produce a prediction for a mouth on the column on the right and the prediction for a nose on the column on the left. And to do that, it has to do the coordinate transforms and that top-down network has to coordinate with the transforms, which may not sound that difficult. And that is not difficult in graphics. In perception, they're quite com complicated because of uncertainty and dealing with probability distribution is much trickier. I am about to put a paper on archive explaining how neural networks can do across that. And that is what that has to do and that is implicit functions and my third big advance and this is revolutionizing the graphics. Gives you the ability to have high-equality computer graphics that learns. All of that is going out of the window of implicit functions. Okay. I talked you through that with the little island example. You have to know where the like is in the intrickic form of reference in the whole thing and then you can say what part can be expressed that. It's like a kidney. A cell has to know which bit of the kidney it is to know which piece of kidney to express. So, how do we form the islands? The idea is that the contextual influence has them settling down in this process and it's trying to be similar to the imbedding victors at the nearby levels and it's trying to be similarom to similar imbedding vectors and in traformers, you converted it to a key and a query and a value. Here, we take the imbedding and don't convert it to anything. You look at the level L embedding vector of like X. You take the scale of product with the imbedding vector at a nearby location, Y and Y and you weight the scaled of products for all of the things in the context and normalize them. The sum of the effects is one. And no you

you're going to pay a lot of attention to similar embedding vectors and little attention to very different embedding vectors and that is how you decide contextual influence. After that, you decide the levels you get overlocations, the echo chambers and that will settle down. I should stress this system is just, I have all sorts of intuitions of the engineering things like this and about how to make these things work. I find it easier to express my intuitions by having an imaginary system and telling you how it works and Hopefully, I have made some things work in the past will make you at least tend to believe I am not just doing philosophy here. I believe this can be engineered to work and things like it that already work. If ask you how to train this thing, the obvious way to train it is with deep training. What you -- deep end-to-end training. You take an image and remove some patches from it. So you would mask the patches and you would let the network settle down. And at the end of settling, you would have to fill in the removed points. You remove words. After all of the layers embeddings, it has to reconstruct the missing words. So you can do the same thing here. You can let this network settle down. And then it has to fill in missing pieces. You get to train everything. That is the easy way. For now, that will do for this technology. The extra term making it different from BERT, is in addition to training it to be able to fill an image, image patches, you put a regularizer to Encourage the Islands of agree. The regularizers are simple. It's like regularizer you can trust in learning. Let me go back to the Island picture. This picture. These are different locations now. If you consider the black arrows at the top, these locations are Talking to each other. You would like them to agree and what is going to happen is? And -- that is more there. Yeah. Let's take the rightmost level L here. That level L is getting three influences from within the same location and the bottom-up net, making up for the larger thing you find in level L. The top arrow, the red arrow, you make the prediction safe from the face to the mouth. And from other locations, you are getting a prediction from the other locations that is a contextual prediction to find that you agree with the other nearby locations. And so, the way you can train that blue arrow and that red arrow, the bottom-up and top-down neural nets is to say, okay, they make a prediction. We have something that is a better opinion than that and that is the consensus of what is happening in the top-down and bottom-up says, and the neighboring location says. You take the consensus opinion from all of the information and you treat that as a supervision sequel. You say when I combine all of these sorts of information against something I believe more than anyone saw, and so I can use that to train the individual sources. It's basic. What you do is you have a prior and you combine the prior and likelihood to get a posterior and train on the prior to get the posterior. And you say the combined thing, saying to train my prior to match my posterior. You're going to train that top-down red model and bottom-up blue model to be a closer fit to the 52 steer are you get when you take both models into account and neighboring locations. When you do that training, you are going to be encouraging in neighboring locations to agree. So that regularizer is going to Encourage you to form Islands. I'm almost done.The fun thing I wanted to address is isn't it terribly -- to replicate. What you get from programming a computer with RAM is if you have the same thing, you want to represent one place and you don't want to replicate all over the place. That, notice that is not what biology does. Biology is happy to replicate if it's convenient. And the reason this replication is very convenient is after you have settled down on objects, thundershower the same and you could have represented with one thing. If you're in the process of segmenting the image and haven't decided what echo chamber to join, you want to have ambiguous representing as and say I am sort of like this and that and for the time being, I like both of them and see how things work out. I will try to be like both of them and during the search for how to segment the image, how to form the positive itry, it's helpful to have the highly redundant represent Augusts and with every like -- representations and with every location, you will have a bet. A whole vector and about -- [Inaudible] All of these things are going to be interactive and making the vectoras the locations or interact whiff each other and that is not like finding a cluster. The data is changing. I will give you data points and you have to find clusters. Here, you are trying to find clusters, the Islands and they're composed of things that keep changing. And the normal idea, let's have the memes and try to move it around to have data doesn't work here. The data is evolving in complicated ways and a much Peter way is at each like to have a vector and let those vectors sort of argue with each other and agree to disagree so you get the different ecochambers. You agree with people there. Um, so I think this is going to be a better way of finding clusters in data that is changing than, for example, the -- [Inaudible] For gaussians. Now, the other thing this does for you is you would like long-range interacts with an image but not complications. Already in the long-range transformers, if they want long-range interactions, they want sparse. They're missing out and not looking at things but sampling. That is fine here. You can have short-range interactions at low levels where parts are small and higher levels, where you get big islands, you can Ford to have sparse long-range interactions, particularly if they're different. The long-range interactions need to sample one point in an island to get all of the information about that thing. Each of these imbedding vectors contains all of the information about that face. The expression and pose. Who it is and that kind of stuff. It's in every vector. So you need to just sample one of them to have that. In summary, I explained three important advances in neural networks, transformers, SimCLR and implicit functions. I showed how you can combine those to produce a multilevel representation of what is going on at each image patch and this solves the problem of how you get parse trees into mural nets. Now, as far as I know, I will -- neural in thes. As far as I know, I am the only person working on this problem. Nobody is interested. The other neuroscientists are -- well, I will leave it at that. Nobody sells worried by the problem that you can't allocate neurons and therefore, how do you represent parse trees? What I size very complicated but there is going to be a long rambling 40-page paper quite soon. Now I think I am done. Yeah. Okay. I'm done.

With that, thank you very much. Great talk. What I really love is, as you said, the idea is very intuitive once you grasp it. Hope many people will be interested in pursuing the topic and representing a park hold, whole park trees. What is going to happen now is we have time until 12:30 for question-and-answer. Reminding people from National Science Foundation, at 3:00, we have an office hour just for our program officers. So, type in to the the question-and-answering Q&A box, a thumbs up to upload questions and I will begin with our most popular question. Why should we aim for human perception when humans have clear deficiencies such as in very sparse and long-range dependencies like those seen in genetics and epigenetics. More general, of course, humans fall prey to all kinds of optical illusions. So, is, is the motivate here to understand the other human system or understand the practical human.

My primary motivation is to figure out how the brain works. I haven't done well with that so far. But this being a side affect of trying to do that, which is various engineering suggestions come out of attempts to figure out how the brain works. It turns out some of the engineering suggestions have been rather successful. So, I am really a fake. I want to figure out how the brain works. In my spare time, I pretend to be an engineer. That is the way on an next to figure out how the brain works. And so, that is my motivation. Right? I mean it could be worse. In the middle ages if you wanted to do research, had to be a monk. Now, you pretend to be an engineer. That is my motivation. But let me give you a very good reason for trying to figure out how to make neural nets be more like people. You don't want to be surprised by what they do. So, adversarial images are really amazing. They're really surprising. I can take an image of a Panda and change the pixels in perceptively and my convoolution neural net will be convinced it's an ostrich. In fact, I can probably take an image of a stop sign and convince it that it's a neural image picture of an ostrich, too. That is surprising and a strong indication whatever convolution they doing is not what we're doing. That is the big difference. Maybe they're doing what we're doing and small dealing with transformers implicitly. A lot of what they're doing is very sophisticated texture discriminations. Much more sophisticated than us. And you can mess these with adversary examples. If they were working more like us, they could not take something like a Panda to us and so it as an ostrich. So, adversary examples should give you a strong reason to make them more like people. We would like to be able to trust them a bit more. If they were more like people, at least they can go wrong the same ways people do and we'll get some insight into what might confuse them. Until they work more like people, you will have no clue what might confuse them, unless you do really extensive testing. Even with really extensive testing, there is going to be the weird cases like an overturned white truck on the highway which just looks like sky to the Tesla. And, yeah. So that -- I mean that is my honest motivation of wanting to know how the brain works and the actual practical motivation is you want to be able to trust them.

Absolutely. Of course, this is the big fear with people who are pursuing end-to-end deep learning for autonomous vehicles with just the standard multilayer that they fall victim so easily to. Optical illusions and we can let humans. We might be able to put up with the kinds of optical illusions that humans suffer from. The ones that are really disastrous.

If Tesla got the same optical illusions as I get, I would be much more confident that we're seeing things the same way as me.

Just a quick question, what does GLOM stand for?

G-l-o-m?

There is a word called gloom it, right?

Oh.

Sticking things together, called glomerate and a popular slang in English called GLOM things together. This is GLOM things together.

Yes. Okay.

Now, some people said it might stand for geoff's last original model. I don't like that. That looks silly to me. And it looks silly to me is LSTM.

Okay. There is a question on the list that I think brings up an interesting issue where it said on Halloween, I wear a mask with two noses. And in general, people can adapt very quickly to novel things that sort of break our high-level expectations. It may take us a minute but we -- and how GLOM handles the novel objects.

Yes. And remember, this is like talking about characters in a novel. GLOM doesn't actually exist. You can't -- characters in a novel that are problematic. Now GLOM handles things is problematic. Nevertheless, I have proposals for it. And so, this one issue is what I call fast weights. So the neural nets, we have two time scales, typically. You have the activity time scale where things change rapidly in an image. And they have the weights where if you are given an image, nothing changes. They slowed it with learning. You obviously won't have things on that time scale, which is weights that adapt rapidly. They can be useful to things like short-term memory, working memory. If you want to implement a reclusive function in the neural network, what it will mean to have a reclusive funk is that since the -- function is that since the knowledge and connects, you get to use the same neurons and connects. -- connections. Let's suppose that painted on your nose I have a face. Right. I look at your face and in small detail. This is not realistic. Reflected in your glasses, I have a face. So I look at your face and I look at the detail and that is a face. So I have this recursive call and I want to use the same knowledge and better use the same neurons. That is where the knowledge is. If I am going to do that, I have to be able to then pop back to the whole face. I have to be able to do the, take things off of the stack again that I put on the stack. Where is that stack? Well, in neural nets, you can implement the fast weights. You have 10 preoverladen weights that allow you to do a fast retrieval of things recently. That is the first neural net model I made. The fast talk navigate in 1973 about how to do reconsidersive neuter -- cursive -- recursive neural weights and it's coming into fashion. I will wait another year before I reimplement it and it will be 50 years since I first did it.

That -- [Inaudible]

Fascinating and I said, I will look forward. A quick followup. So, the two-part question. Is there biological evidence of some implementation of fast weights and then how do the fast weights get turned into long-term memory? The brain involves something like the hippo campus and sleep and all sorts of things.

Right. There are all kinds of things. So, come back to sleep in a minute. But if you talk to neuroscientists about where temporary memories are, well, the neuroscientists I talk to, in fact, the first time I talked to [Inaudible] , he was concept to educate me that there is not one time scale in synaptic changes. He was working on frogs at Harvard then and he said, there are fast sippantic changes and less fast and your model needs to contain two things. You shouldn't have a one-time scale. Yes, there are lots of neural evidence. I talked to Simon [Inaudible] Who did a Ph.D recently at M.I.T. and you should look for when memory is for things that are a minute old and you could have a bunch of neurons going around going ping, ping, ping and remembering stuff or adapt sin an simp. Seems the temporary adaptation is a sin anes is. For example, if I say, the word sleep now, you will recognize that 30 seconds ago. Where is that storm and knowledge that you mentioned 30 seconds ago? Do you really have neural activity sitting around remembering that? Or did you just temporary change the sinnance strength involved in the word sleep and associations so that the sleeping noise 30 secs later, I will be better at recognizing it and seems like it's temporary changes. The essential answer is temporary weight changes.

Great. Another question. I think from people who think a lot about medical imaging. How does -- and the example like the eyes are here and the nose is down below. So, in medical where people are looking at, like images of a cluster of cells and they're trying to find out if cells are cancerous or not, but they don't have the geo metric distribution. Does that also still fit within the whole hierarchy?

I think it's going to be the case. I'm not an expert of images. I have been quoted as saying we need to do better and I believe that still. For example, if I look at a bunch of cells in a pathology slide, I might expect to find a bunch of cancer cells next to one another. I think they tend to replicate. In fact, that is what they tend to do rather well. Forever, you tend to get a bunch nearby each other. So, that is fitting with what I get. You expect to segment it into similar things being neuron andother.

We have career-related questions and they let us summarize that -- give advice to -- and other lectures. Say someone is listening to this talk and is an undergraduate and maybe they have been working for a few years. And they say, oh, this sounds great. What should their next steps be? Grans in neuroscience? What do you recommend people do?

I have standard advice I give, which is about troughing your intuitions. Which does not answer this question. I ask can answer different questions.

Okay.

And how should I organize my career. You should trust yourp tuitions and in particular, you should focus on where you have a strong intuition that everyone is doing it wrong and you don't know how to do it right yet, but there is something fishy about whether or not everyone does is doing and there is something fishing -- fishy. What the brain does is not that. It allocates the locations and then allocates the vector to represent the nose and that is an activity vector. You can create those on-the-fly. And really trust your endueigs when you think what everyone says is fishy and focus on that. It's taken me like 40 years to figure out how to make this work and just keep focusing on that. You either have good indueitions or you don't. And if you have good indueitions, trust them. If you don't, it doesn't matter what you do. You might as well focus on your intuitions.

And there is Auld the issue of not everyone starts out with the great intuition.

And that helps an adviser to Encourage and draw her intuitions.

And I will answer the question. What you should do is fit yourself a good background in linear aljeb bra, computer science and everything else, neuroscience, all of those things and then get yourself into a graduate school at a really good place where you will be, you will have an adviser who is smart and more importantly, you will have other grad students to talk to and that is not an incow baiter, if they came up with stuff, that is done.

Uh-huh.

And obviously you should do that and that is a waste of time unless you have strong indueigs of your own and you trust them.

And that is great. Some more questions here and contemporary language models like BERT and Gp 3 include a lot of data and are hard to replicate unless you're an enormous company. And I, is the same thing going to apply to GLOM and my intuition, see if that is correct, when you're looking at a radically new approach, often you can explore that at a smaller scale. You don't necessarily need to learn a trillion system. And what is your answer?

My answer is I think it's similar in physics and for example, the experiment that proves quantum mechanics is right and one goes through both slits. And that does not require a fancy aprateus. It requires billions and billions of dollars in apparatus. There are some things you can do on a small scale and others you can't. I believe the biggest advances will come from things you do on a small scale. And I do small scale stuff myself. I believe they can be explored on the big scale. And so I am, I sit on the fence. I believe in doing small-scale scientific stuff to understand new ideas and solve basic problems that are not solved yet. And I believe that we learn a lot from the huge suspicious and whether you go from a billion parameters to a trillion, things get better and I don't have patience saying you shouldn't be allowed to publish the papers. We can't represent that. We don't have, we can't use kind of a million years of cpu time to replicate this experiment. And they didn't say to publish pagers, we can't have that in the backyard and so, there are some things the big companies can do and places like deep mind and face book and so on. Luckily a bump of them and that I can check each other. We have to be open to that kind of research. And we have to be willing to publish that, even though it can't be replicated by people and -- in the computer science department. We'll get bigger wins in universities and where you're exploring a bigger diversity of ideas.

Okay. And have I have a question,modification of the questions here. Let's say that GLOM is right and you're a neurotist. What should you be looking for in the brain that might, that would help confirm that approach?

I am working on two papers. I am going to put one in the archive in a week or two. And there is another paper that has work to be done. That is going took make a lot of predictions and.

Excellent. I can say one thing. It might be interesting to neuro tists. -- scientists. If you look at the theories of sleep, the standard neural net sleep repressant is that you drive things down from the hippo campus and it allows you to do rehearsal and so you can reverse during the day. It allows you to you want great data during the day with older data and so they fit in the same bottle. So you're not overwriting the old or new experience. You make sure that both work and that is one serious sleep. It's not satisfactory. You, if I deprive you of sleep for a few days, you go psychotic and if I deprive you of sleep for a week, you may neverrecover and here's a theory we cooked up recently. You're doing this contrastive learning where you want two patches of the same image to have the same representation and you wanted to images to have different representing as. Contrastive learning involves a physical phase where things that should be the same remain the same and a negative phase, like a partition function, where things that are different should be kept different. And things that shouldn't be confused are kept different. Seems hard to run both of those faces as once when the data is coming in and it's clear when it's coming in, you could actually be making things that should be the same more similar. That is you could be at each level and get the predictions and make it green more with the contextual predictions and vice-versa and you can do that positive face where you're awake. What about the negative phase and making things different different? And the very best negative examples are from the same video and on the adjacent frames and a few frames away. And you can imagine a window to make your representing a of this frame be similar to the representing as of neighboring frames and dissimilar from those ripped further away. Those are the ones you would get confused by. And that way you will force yourself to get smooth representing as on that variance. And this is how you can do yet in the brain. You can say during the day, you make things more similar. You don't worry about holding them apart. During the day, the representing as will slowly collapse towards each other. And now, you go to sleep and you look for things that are now too similar and should be different. The way do you that is you would yate a video and look at them further away. If you have a particular window that you're learning and making things similar or dissimilar, you could look for things further away by playing the video faster. And in sleep what you might do is recreate a video and make it go seven times as fast. You can go do that with GLOM. You have the top-down model and the top-down model doesn't require any attention. Attention to neighboring things is required for contextual disambitioniation. You have uncertainty. When there is no uncertainty and you can generate top-down very fast. From the top level, I run my video rapidly. And from that, I generate all of the other levels and at high speed. And then I do unlearning. The opposite of learning. Just opposite of what I did during the day and that is taking the hard negative examples and that I should be moved into the window and if you decided to try and make ground compatible with biology, there are two big questions. And will it tolerate having a lot of positive learning? The answer is yes and that is tolerant of that. Things start collapsing. And they can collapse before considerable degeneration. And you can do a lot of negative learning and that is capable of dealing with a day's worth of learning and unlearning and the second question, is there any evidence at all? When you first go to sleep, you have a face of sleep with the spindles and you're replaying the day's experience. Seven times as fast. They see this in rats. Rats do it during sleep and after they have been through a maze. And you can record from these place fields to follow a trajectory. During sleep, they do the same trajectories and sometimes fast. They also may replay things backwards and that is another story and that is 0ing. There is evidence that the spindle phase of sleep is for replaying things fast and for contrastive learning, that would fit in and that would explain why that phase of sleep causes psychosis. Direct representing as collapse. And -- representations collapse and you get totally confused.

And thank you. That is great. Your answer to the question is like worth.

Telling you?

-- .

that is wonderful. It's like a whole in Mia notes here, another pocket. And I look Ford to that. That addresses some issues of sleep and why we sleep, a popular account of sleep CBS and he concentrates on the unconscious faces of sleep that involve a lot of cleanup of e poxins and so forth as well.

I am sure there are lots of other things. It does lots of things, right?

I was not aware of that spindle. That explains why it seems like -- for hours. And only a few minutes, really, past really time. And we're going to have to draw to a slows there. Unless you handle -- to a close there. My last question was going to be to ask you to talk about Kent.

Talk more about what, sorry?

You mentioned the influence of a critique of pure reason of theory and time and space. And if you want to take two minutes of how that influenced your thinking?

Yeah. There was something that always intrigued me in that which is he said, something like -- this is amateur philosophy. And space is the form of the external sense and time is the form of the internal sense and that just intrigued me whether I read it as a Phils on fit fify student and that welled to me learning about objective funks for learning, and in particular, the special colearns, a different adjective function from what to -- and also temporal coherence.

Okay, great. Thank you so much. Thank our audience and thank our tech crew and we're always thinking about our research portfolio and what areas we should be doing more to promote. I think you have given us a lot of --

And can we make one idea?

Yes.

And the deep learning revolution. A lot of it came from me and [Inaudible] In Canada.

Yup.

And 1$00,000 a year for a whole lab. And they were unrestricted. Ef five years, you write a proposal and there are six pages. Nobody holds you to it and that kind of support for basic research is really, really valuable in the very long run. It's very hard to sell to politicians and we're funding them to do what is cute. Doesn't sound like a good funding proposal, and the really big payoffs come from that. Canada has a good program there. It doesn't have much money for research or put Nearly enough in, but one thing it does right is the research grants. They go to a wide range of people. You can't pick winners in advance. You can give them to smart people and use peer review, but you need to distribute it fairly broadly, o posed to putting the opposite of this that would be the European blue brain project, whatever it's called and where you put a billion dollars into the crazy peer -- and experiments and results. A waste of a billion dollars.

And let's talk about this when we're in the NSF session this afternoon. Thank you, again, and thank you to the audience and to everyone else. Very exciting morning. Take care, everyone.

Okay. Bye.

Bye. [Event concluded]