The following content is provided under a Creative Commons license. Your support will help MIT OpenCourseWare continue to offer high quality educational resources for free. To make a donation or view additional materials from hundreds of MIT courses, visit MIT OpenCourseWare at ocw.mit.edu.

TOMASO POGGIO: So I'll speak about i-theory, visual cortex, deep learning networks. The background for this is this conceptual framework that we take as a guide to present work in vision in this center-- The idea that you have a phase in visual perception, essentially up to the first saccade-- say, 100 milliseconds from onset of an image-- in which most of the processing is feedforward in the visual cortex. And that top-down signals-- I hate the term feedback in this case, but back projections going from higher visual areas, like inferotemporal cortex, back to V2 and other cortical areas are not active in this first hundred milliseconds.

Now, all of this is a conjecture based on a number of data. So it has to be proven. For us it's just a motivation, a guide, to first studying feedforward processing in, as I said, the first 100 milliseconds or so. And to think that other types of theory, like generative models, probabilistic inference that you have heard about, visual routines you have heard kind of from Shimon, are important not so much in the first 100 milliseconds, but later on. Especially when feedback through back projection, but also through movements of the eyes that acquire new images depending on the first one you have seen, come into play.

OK. This is just to motivate feedforward. And of course, the evidence I refer to is evidence like-- you have heard from Jim DiCarlo, for the physiology there is quite a bit of data showing that neurons in IT become active and selective for what is in the image about 80 or 90 milliseconds after onset of the stimulus. And this basically implies that there are no big feedback loops from one area to another one. It takes 40 milliseconds to get to V1, and 10 milliseconds or so for each of the next areas.

So the problem is, computational vision-- the guy on the left is David Marr. And here it's really where most probably a lot of object recognition takes place, is the ventral stream from V1 to V2, V4, and the IT complex.

So that's the back of the head. As I said, it takes 40 milliseconds for electrical signals to come from the eye in the front through the LGN back to neurons in V1. Simple complex cells. And

then for signals to go from the back to the front, that's the feedforward part.

And on the bottom right, you have seen this picture already. This is from Van Essen, edited recently by Movshon. It's the size of the areas and the size of the connection are roughly proportional to the number of neurons and fibers. So you see that V1 is as big as V2. they both have about 200 million neurons. And V4 is about 50 million, and the inferotemporal complex is probably 100 million or so.

Our brain is about one million flies. A fly is around 300,000 neurons or so. A bee is one million. And as I think Jim DiCarlo mentioned, there are these models that have been developed since Hubel and Wiesel-- so that's '59-- that tried to model feedforward processing from V1 to IT. And they start with simple and complex cells, this S1 and C1, simple cells being essentially equivalent to Gabor filters, oriented Gabor filters in different positions, different orientations.

And then complex cells that put together the signals from simple cells of the same orientation preference, but different position, and so have some more position tolerance than simple cells. And then a repetition of this basic scheme, with S2 cells that are representing more complex-- let's call them features-- than lines. Maybe a combination of lines. And then C2 cells again pulling together cells of the same preference in order to get more invariance to position.

And there is evidence from the old work of Hubel and Wiesel about simple and complex cells in V1. So S1 and C1, although the morphological identity of complex and simple cells is still an open question-- you know, which specific cells. We can discuss that later. But for the rest, this hierarchy continuing in other areas, like V2 and V4 and IT, this is one conjecture in model like this.

And we, like other ones before us, modeled back 15 years ago this different area. It's V1, V2, and V4 with this kind of model. And the reason to do so was not really to do object recognition, but it was to try to see whether we could get the physiological properties of a different sense in such a feedforward model, the ones that people have had recorded from and published about. And we could do that to reproduce the property.

Of course, some of them we put in properties of simple and complex cells. But other ones, like how much invariance to position there was in the top level, we got it out from the model consistent with the data.

One surprising thing that we had with this model was that, although it was not designed in

order to perform well at object recognition, it did actually work pretty well. So the kind of things you have to think about this is rapid categorization. You have seen that already. And the task is, for each image, is there an animal or not? And you can kind of get the feeling that you can do that.

In the real experiment, you have an image and then a mask, another image. And then you can say yes, there is an image, or no, there is not. This is called rapid categorization. It was introduced by Molly Potter, and more recently Simon Thorpe in France used it. And it's a way to force the observer to work in a feedforward mode, because you don't have the time to move your eyes, to fixate. There is some evidence that the mask may stop the back projections from working.

So this is a situation in which you could compare human performance to these feedforward models, which are not a complete description of vision anyway, because they don't take into account different eye fixation and feedbacks and higher processes, like-- like I said, probabilistic inference and routines. Whatever it happens, very likely in normal vision, in which you have time to look around.

So in this case, this d prime is a measure of performance, how well you're doing this task. And you can see, first of all, the absolute performance, 80% correct on a certain database. This task, animal no animal, is similar between the model in humans. And images that are difficult for people, like images in which there is a lot of clutter, the animals are small, are also difficult for the model. And the easy ones are easy for both. So there is a correlation between models and humans.

This does not say that the model is correct, of course, but it gives a hint that model of this type capture something of what's going on in the visual pathway. And Jim DiCarlo spoke about a more sophisticated version of these feedforward models, including training with back propagation, that gives pretty good results also in terms of agreement between neurons and units in the model.

So the question is why these models work. They're very simple, feedforward. It has been surprisingly difficult to understand why they work as well as they do. When I started to work on this kind of things 15 years ago, I thought this kind of architecture would not work. But then they worked much better than I thought.

And if you believe deep learning these days, which I do-- for instance, in performance on

ImageNet-- my guess is they work better than humans, actually, because the right comparison for humans on ImageNet would be the rapid categorization one. So they present images briefly. Because that's what the models have-- just one image. No chance of getting a second view.

Anyway, that's a more complex discussion that has to do also with how to model the fact that in our eyes, in our cortex, every-- solution depends on eccentricity. It's a pretty rapidly decaying resolution as you go away from the fovea, and has some significant implications for all these topics. I'll get to that.

What I want to do today is, one way to look at this to try to understand how these kind of feedforward models work-- i-theory is based on trying to understand how models that are simple and complex cells and can be integrated in a hierarchical architecture can provide a signature set of features that are invariant to transformations observed during development, and at the same time keep selectivity. You don't lose any selectivity to different objects.

And then I want to see what they say about deep convolutional learning networks, and look at some of the-- beginning with theory about deep learning. And then I want to look at a couple of predictions, particularly related to eccentricity-dependent resolution coming from i-theory, that are interesting for the sake of physics and modeling. And then it's basically garbage time, if you're interested in mathematical details and proofs of theorems and historical background.

OK. Let's start with i-theory. These are the kind of things that we want, ideally, to explain. This is the visual cortex on the left. Models like HMAX, or feedforward models. And on the right are the deep learning convolutional networks, a couple of them, which basically have convolutional stage stages very similar to S1, and pooling stages similar to C1. But quite a lot of those layers.

How many of you know about deep learning? Everybody, right? OK. These are the kind of questions that i-theory tries to answer-- why these hierarchies work well, what is really visual cortex, what is the goal of V1 to IT. We know a lot about simple and complex cells, but again, what is the computational goal of these simple and complex cells? Why do we have Gabor tuning in the early areas? And why do we have quite generic tuning, like in the first visual area, but quite specific tuning to different types of objects like faces and bodies higher up?

The main hypothesis with starting i-theory is that one of the main goals of the visual cortex-- it's a hypothesis-- is to compute a set of features, a representation of images, that is invariant

to transformations that the organism has experienced-- visual transformations-- and remains selective.

Now, why is invariance important? A lot of the problem of recognizing objects is the fact that I can see once Rosalie's face, and then the next time it's the same face, but the image is completely different, for it's much bigger now because I'm closer, or the illumination is different. So the pixels are different. And from one single object, you can produce in this way-- through translation, scaling, different illumination, viewpoint-- you can produce thousands of different images.

So the intuition is that if I could get a computer description-- say, long vectors of features of her face-- that does not change under these transformations, recognition would be much easier. Easier means, especially, that I could learn to recognize an object with much fewer labeled examples.

Here on the right you have a very simple demonstration of what I mean, empirical demonstration. So we have at the bottom different cars and different planes. And there is a linear classifier which is trained directly on the pixel. Very stupid classifier. And you train it with one car and one plane-- this is on the left-- or two cars, two planes. And then you test on other images.

And as you can see, when it's trained with the bottom examples, which are at all kinds of viewpoints and sizes, the performance of the classifier in answering is this a car or is this a plane, it's 50%. It's chance. Does not learn at all.

On the other hand, suppose I have an oracle which is-- I will conjecture is visual cortex, essentially, that gives you the feature vectors for each image, which is invariant to these transformations. So it's like having images of cars in this line B. They're all in the same position, same illumination, and so on, and the same for the planes.

And I repeat this experiment. I use one pair-- one car, one plane-- to train, or two cars, two planes, and I see immediately that when tested on new images, this classifier is close to 90%. So much better. So correcting-- having invariant representation can help a lot.

That's the empirical, simple demonstration. And you can prove theorems saying the same thing, that if you have an invariant representation, you can have a much lower simple complexity, which means you need much fewer labeled examples to train a classifier to

achieve a certain level of accuracy.

So how can you compute an invariant representation? There are many ways to do it. But I'll describe to you one which I think is attractive, because it's neurophysiologically very plausible. The basic assumption I'm making here is that neurons are very slow devices. They don't do well a lot of things.

One of the things they do probably best is high-dimensional dot products. And the reason is that you have a dendritic tree, and in cortical neurons you have between 1,000 and 10,000 synapses. So you have between 1,000 and 10,000 inputs. And each input gets essentially multiplied by the weight of the synapse, which can be changed during learning. It's plastic. And then the post-synaptic depolarization or hyperpolarization, so the electrical changes to the synapses, get all summated in the soma.

So you have some i. Xi are your inputs, Wi are your synapses. That's a dot product. And this happens automatically, within a millisecond. So it's one of the few things that neurons do well. It's, I think, one of the distinctive features of neurons of the brain relative to our electronic components, that in each neuron, each unit in the brain, there are about 10,000 wires getting in or out. When I say in, transistor or logical units in our computers, the number of wires is more like three or four.

So this is the assumption, that this kind of dot products are easy to do. And so this suggests this kind of algorithm for computing invariance. Suppose you are a baby in the cradle. You're playing with a toy-- it's a bike-- and you are rotating it, for instance. For simplicity. We'll do more complex things.

The unsupervised learning that you need to do at this point is just to store the movie of what happens to your toy. For instance, suppose you get a perfect rotation. This is a movie up there. There are eight frames. Yeah. You store those, and you keep them forever.

All right. So when you see a new image, it could be Rosalie's face, or this fish. And I want to compute a feature vectors which is invariant to rotation, even if I've never seen the fish rotated. What I do is, I compute a dot product of the image of the fish with each one of the frames. So I get eight numbers.

And the claim is that these eight numbers-- not their order, but the numbers-- are invariant to rotation of the fish. So if I see the fish now in a different rotation angle-- suppose it's vertical,

I'd still get the same eight numbers. In a different order, probably. You could have-- these are eight numbers. What I said, they are invariant to rotation of the fish.

There are various quantities that you can use to represent compactly the fact that they are the same independent of rotation. For instance, the probability distribution-- the histogram-- of these values does not depend on the order. And so if you make a histogram, these should be independent of rotation, invariant to rotation, Or moments of the histogram, like the average, the variance, the moment of order infinity.

And for instance, the equation for computing a histogram is written there. You have the dot product of the image, the fish, with one template to the bike, the bike Tk. You have several templates, not just one. And Gi is the element of the rotation group. So you get various rotations of-- simply because you have observed that. You don't need to know its rotation group. You don't need to compute that. These are just images that you have stored.

And there can be different thresholds of simple cells. And sigma could be just a threshold function, for instance. As it turns out-- I'll describe later. And sum is the pool. I'll describe later these. But sigma, the nonlinearities can be, in fact, almost anything. This is very robust to different choices of the nonlinearity and the pooling.

Here are some examples in which now the transformation is translation that you have observed for the bike. And if I compute a histogram-- from more than eight frames, in this case-- I get the red histogram for the fish, and you can see the red histogram does not change, even if the image of the fish is translated.

Same for the blue Instagram, which is the set to features corresponding to the cat. Also it's invariant to translation. But it's different from the red one. So these quantities, the histograms, can be invariant of course, but also selective, which is what you want.

In order to have a selectivity as high as you want, you need more than one template. And some results about how many you need. I can go into more details of this. But essentially, you need a number of templates-- of templates like the bike, in your original example-- that is logarithmic in the number of images you want to separate. For instance, suppose you want to be able to distinguish 1,000 faces, or 1,000 objects. Then the number of templates you need is in the order of log 1,000. So does not increase so much.

Yeah. So there are two things, one, which you implied. The reason I spoke about rotation of

the image plane, because rotation is a compact group. So you never get out. You come back in. The translation, you can-- in principle, mathematically, between plus infinity or minus infinity. Of course it does not make sense, but mathematically this means that it's a little bit more difficult to prove the same results in the case of translation and scale. But we can do it. That's the first point.

The second one, the combinatorics of different transformations. Turns out that-- one approach to this is to have what the visual system seems to have, in which you have relatively small ranges of invariance at different stages. So that at first stage, say in V1, you have pooling by the complex cells over a small range of translations, and probably scale. And then at the second stage you have a larger range. I'll come to that later. But it's a very interesting point.

I'll not go into this. These are-- technical extension of these partial observer groups, these non-compact groups. The non-group transformation of this approximate invariance to rotations in 3D, or changes of expression, and so on. And then what happens when you have-- a hierarchy of just modules. I'll say briefly something about each one.

One is that if you look at the templates that give you simultaneous-- so what we want to do, we want to get scale and positioning invariance. And suppose you want templates that maximize the simultaneous range of invariance to scale and position. It turns out that Gabor templates, Gabor filters, are the ones to do that. So that may be one computational reason for why Gabor filters are a good thing to do in processing images.

So for getting approximately good invariance to non-group transformations, you need to have some conditions. The main one is that the template must transform in a similar way to the object you are to compute, like faces. And for these properties to be true for a hierarchy of modules.

Think of this inverted triangle like a set of simple cells at the base, and one complex cell, the red circle at the top. And so the architecture that we're looking at is simple complex. This would be like V1. And next to it, another simple complex module. This is all V2. And then you have V1 in the second layer, that is getting the input from V1. And you repeat the same thing, but on the output of V1.

This is exactly like a deep learning network. It's like visual cortex, where you have different stages and the effective receptive fields increases as you go up, as you see here. So this would be the increase in spatial pooling-- so invariance-- and also, as I mentioned-- not drawn

here, but the scale. Pooling over size, scale. And you can show that, if the following is true, that-- let me see. Is this animated? No.

What you need to have-- and a number of different networks, certainly the ones I described, have this property of covariance. So suppose you have an object that translates in the image. OK. What I need is that the neural activity-- the red circles at the first level-- also translate. This is covariance.

So what happens is the following. Suppose the object is smaller than those receptive fields, and this drawing is as big. But suppose it's smaller. Then if you translate one of those receptive fields, going from one point to another, because each one has invariance to translations within the receptive field-- it's pooling over them-- translation in the receptive field will give the same output. You will have invariance right there.

But suppose you have one image, and then the next one the object moves to a different receptive field, or gets out of the receptive field. Then you don't have invariance at the first layer. But if you have covariance-- or the neural activity moves-- at that layer above, you may have invariance under that receptive field. In other words, in this construction, if you have this covariance property, then at some point in the network, one of these receptive fields will be invariant.

Is that--

AUDIENCE: Can you explain that again?

TOMASO POGGIO: Yeah. The argument is-- suppose I have an object like this. I have an image. And then-- I have another image in which the object is here. Obviously the response at this level-- the response of this cell will change, because before it saw this object. Now, there is these other cells who see that. So the response has changed. You don't have invariance.

However, if you look at what happens, say, at the top red circle there, the top red circle will you see some activity in the first image here, because it was activated for this. And-- in the second case, we see some activity over there, which should be equivalent. And under these receptive fields, translations will give rise to the same signature. Under this big receptive field, you have invariance for translation within it.

So the argument is that-- either you have invariance at one layer, because the object just

moved within it, and then you are done. It's invariant, and everything else is invariant. Or you don't have invariance in this layer, but you will have it at some layer above. So in a sense-- if you go back to this-- I'll make this point later. But if you go back to this-- to this algorithm, the basic idea is that you want to have invariance to rotation. And so you average over the rotations.

But suppose you want to have invariance-- you want to have an estimate of rotation, but you're not interested in identity. Then what you do, you don't pool over rotation. You pull over different objects at one rotation. So you can do both. All right?

AUDIENCE: My question was more physiological than theoretical.

TOMASO POGGIO: Yeah. Physiological-- we had done experiments long ago in IT with Jim DiCarlo, Gabriel Kreiman. And from the same population of neurons, we could read out identity, object identity, invariant to scale and position. And we could also read out position invariant to identity. And--

AUDIENCE: The same from the--

TOMASO POGGIO: Same population. I'm not saying the same neuron, but the same population of 200 neurons. And so you can imagine that you could have different situations. One could be some of the neurons are only conveying position, and some others are completely invariant. And when you read out with a classifier, it will work. Or you have neurons that are already combining this information, because the channels-- either way.

OK, let me do this, and then we can take a break. I want to make the connection with simple and complex cells. We already mentioned this, but this set of operations, you can think of this sigma dot product, n delta, this is a simple cell.

So this is a dot product of the image with a receptive field of the simple cell. That's what this parenthesis is. You have a bias, or a threshold, and the nonlinearity. Could be the spiking nonlinearity. Could be, as I said, a rectifier. Neurons don't generate negative spikes. And so all of this is very plausible biologically. And the simple cell will simply pool, take the over the different simple cells. So that's what I mentioned before, that nonlinearity can be almost anything.

And I want to mention something that could be interesting for physiology. From the point of view of this algorithm, this may be a solution to this problem that has been around for 30 years or so, which is that Hubel and Wiesel and other physiologists after them identified simple and

complex cells in terms of their physiological properties. They couldn't see from where they are recording.

But there were cells that behaved in different ways. The simple cells had the small receptive field. The complex cell had larger receptive field. The complex cells were more invariant. And then physiologists today are using criteria in which the complex cell is more non-linear than the simple cell.

Now, from the point of view of the theory, the real difference is one is doing the pooling-- the complex cells. The simple cell is not. And the puzzle is that despite these physiological difference, they were never able to say this type of pyramidal cell is simple, and this type of pyramid cell are complex. And part of the reason could be that maybe simple and complex cells are the same cell. So that the operation can be done on the same cell.

If you look at the theory, what may happen is that you have one dendrite play the roll of a simple cell. You have inputs, synaptic weights. So this could give rise, for instance, to the Gabor-like receptive field. And then-- these other dendrites to another simple cell. It's a Gabor-like in a slightly different position in the image plane, in the retina. You need the nonlinearities.

And they may be, instead of the output of the cell, they may be so-called voltage and time dependent conductancies in the dendrites. In the meantime, we know that pyramidal cells in the visual cortex have these nonlinearities like almost having spike generation in the dendrites. And then the soma will summate everything. This is what the complex cell is doing.

And if one of the cells is computing something like an average, which is one of the moments of a distribution, then the nonlinearity will not even be needed. And then physiologists, using the criteria they use this day, would classify that cell as simple, even if from that point of view of the theory it's still complex.

Anyway, that's the proposed machinery that comes from the theory. That's everything that we need. And it will say simple and complex cell could be one cell.