MITOCW | Lec-14

PATRICK WINSTON: So today we're gonna talk about a few miracles of learning in the context of the theme that we're developing here in the class.

We started off with a discussion of some basic methods.

We talked about nearest neighbors.

And we talked about identification trees.

And those are kind of basic things that have been around for a long time.

Still useful.

Still the right things to do when you're faced with a learning problem and you're not sure what method to try.

Then we went on to talk about some naive biological mimicry.

We talked about neural nets.

And we talked about genetic algorithms.

And you look at those things and you think and reflect back on what we talked about.

And you have to say to yourself, are these nugatory ideas?

Perhaps pistareens?

Or are they supererogatory ideas that deserve to be center stage?

Does anybody know what those words mean?

A pistareen?

Well, a pistareen is a Spanish coin.

It was so small.

It was of little worth.

These ideas like neural nets, genetic algorithms, I classify them as pistareens because getting them to do something is rather like getting a dog to walk on its hind legs.

You can make it happen, but they never do it very well.

And you have to think it took a lot of trickery and training to make it happen.

So not too personally high on those ideas.

But we teach them to you anyway because, of course, we only editorialize part of time and part of time we like to cover what's in the field.

Today we're starting a couple of discussions of mechanisms or ideas or things to know about that are quite different.

Because now we're going to focus on the problem rather than on the mechanism.

And then a later on we're going to talk about deep theory, FIOS, for its own sake.

But this week I want to talk about mechanisms that were devised.

I want to talk about research that was done.

Let me not say mechanisms.

Let me say research that was done to attempt an account of some of the things that we humans do well.

Sometimes without even knowing that we do it.

Now Krishna here tells me his first language was Telugu.

Telugu.

I once had another student whose first language was Telugu.

I said to him, that must be one of those obscure Indian languages.

And he said, yes.

It's spoken by 56 million people.

French is spoken by 52.

[LAUGHTER] PATRICK WINSTON: He's going to be our experimental subject.

Krishna, if I pluralize words-- you know what it means to pluralize a word.

So if I say for example, horse, then if I ask you for the plural you'll say horses.

So if I say dog, what's the plural?

STUDENT: Then dogs.

Or in my language?

PATRICK WINSTON: No, no, no.

In English.

STUDENT: Oh, dogs.

PATRICK WINSTON: Well, what about cat?

STUDENT: Cats.

PATRICK WINSTON: And he got it right.

Isn't that a miracle?

When did you start speaking English?

STUDENT: Second grade.

PATRICK WINSTON: Second grade.

But he still got it right.

But he never learned that he's actually pluralizing those words differently.

But he is.

So when you pluralize dog, what's the sound that comes after?

It's a z sound.

Zzzzzz.

Dogzzz.

If you stick your fingers up here you can probably feel your vocal cords vibrating.

If you stick a piece of paper in front of your mouth you'll see it vibrate.

But when you say cats, the pluralizing sound is sss, like that.

No vocalizing.

No vibration of the vocal cords.

And old Krishna here learned that rule, as did all of you other non-native speakers of English, effortlessly and without noticing it.

You learned it.

Buy you always get it right.

How can that possibly be?

Well, be the end of hour you'll know how that might be.

And you'll experience a case study in how questions of that sort can be approached with a sort of engineering point of view.

You can say, what if God were an engineer?

Or alternatively, what if I were God and I am an engineer?

Think about how it might happen that way.

So we want to understand how it might be that the machine could learn rules like that.

Phonological rules.

Not just that one, but all the phonological rules you'd acquire in a course on phonology.

That part of speaking that deals with those syllabic and sub-syllabic sounds.

The phones of the language.

So when Yip and Sussman undertook to solve this engineering problem, both being dedicated engineers, the first thing they did was learn the science.

So they went to sit at the foot of Morris Halle, who would develop-- was largely responsible for the development theories of so-called distinctive features.

And here's how all that works.

You start off with a person who wants to say something.

And out that person's mouth comes some sort of acoustic pressure wave.

And if I say, hello, George.

And you say hello, George.

Everybody will understand that we said the same thing.

But that acoustic waveform won't look anything alike.

It'll be very different for all of us.

So it's a miracle that words can be understood.

In any case, it goes into an ear.

And it's processed.

And out comes a sequence of distinctive feature.

Vectors.

A distinctive feature is a binary variable like is the phone voices or not.

That is to say, are your vocal cords vibrating when you say it?

If so, then that's plus voiced.

If not, it's minus voiced.

So according to the original distinctive feature theory and consistent with most of the theories that have been derived since the original one, there are on the order of 14 of these distinctive features that determine which phone you're saying.

So if you say ah, that's one combination of these binary features.

If you say tuh, that's another combination of these binary features.

14 of them.

So how many sounds does that mean, in principle, there could be in a language?

SEBASTIAN: 2 to the 14th.

PATRICK WINSTON: And what's 2 the 14th, Sebastian?

Well, it ought to be about 16,000, don't you think?

2 to the 10th is 1,000.

2 the fourth is 16.

So there are about 16,000 possible combination.

But no language on Earth has more than 100 phones.

That's strange, isn't it?

Because some of those choices are probably excluded on physical ground.

But most of them are not.

So we could have a lot more phones in our language than we actually do.

English is about 40.

So the sequence of distinctive features could be viewed as then producing meaning after, perhaps, a long series of operations.

But in the end, those operations feedback in here because many of the distinctive features are actually hallucinated.

We think we heard them, but they're not there.

Or they're not even in the acoustic waveform.

They're there for the convenience of the phonologist who make rules out of them.

It's remarkable how much of this feedback there is, and even injection from other modalities.

Many of you may have heard about the McGurk Effect.

Here's who the McGurk Effect works.

Look at me while I say ga, ga, ga, ga, ga, ga.

OK.

I said, g-a.

Now how about ba, ba, ba, ba.

OK.

I said ba like a sheep.

But if I take the sound I make when I say ba and play it while you're taking video of me saying ga, what do you think you hear?

You don't hear ba.

Some people report that they hear a d-a sound like da.

When I look at it, I can't make any sense out of it.

It looks like there's a disconnection between the speech and the video.

But it does not sound like ba.

But if I shut my eyes and say ba, ba, it's absolutely clear that it's b-a.

So what you see has a large influence on what you hear.

It's also interesting-- although a side issue-- it's also interesting to note that it's very difficult pronounced things correctly if you don't see the speaker.

So many people wonder when they learn foreign languages why they can't speak like a native.

And the answer is, they're not watching the mouth of the speaker.

I was talking to a German friend once and said, you know, I just can't say the damned umlaut right.

And he said, oh, the trouble with you Americans is you don't realize that American cows say moo but German cows say muu.

[LAUGHTER] PATRICK WINSTON: And, of course, I got instantly because I could see that the umlaut sounds are produced with protruding lips, which we don't have any sounds an English that require that.

Ah, but back to what we know from the phonologists about all this stuff.

If you talk to Morris Halle, he will tell you that over here-- I like to think of it as a marionette.

There are five pieces of meat down here.

And the combination of distinctive features that you're trying to utter are like the control of a marionette on those five pieces of meat.

So if you want to say an a sound, the marionette control goes into a position that produces that combination.

So let's see.

What does that distinctive feature sequence look like for typical word?

Well, here's a word.

A-e-p-l.

Apples.

And we can talk about what distinctive features are arrayed in that particular combination of phones.

So one of the features that they like to talk about is syllabic.

Syllabic.

That roughly means, can that sound form the sort of core of a syllable?

And the answer is a can, buy these can't.

So it's plus, minus, minus, minus.

Down here a little ways you'll run into the voiced feature.

And for the voiced feature, well, we can do the experiment ourselves.

Ahh.

Sounds like it's voices to me.

Pa.

No.

That's not voiced.

Oo.

Yep.

Zzz.

We already said that was voiced.

So that's the combination you see when you utter apples for the voiced feature.

Then another one is the continuent one.

That roughly says is your vocal apparatus open?

Is there no obstruction?

And so ahh plus pa is constricted.

Oo, open.

Zzz, open.

So that one happens to run right along with voiced in that particular word.

Oh, and there are 14 altogether.

But let me just write down one more.

The strident one.

That says, do you use your tongue to form a little jet of air?

So you don't on aa, pa, oo.

Buy you do on z.

So that gets a plus.

So that's a glimpse through a soda straw of what it would like to represent the word apples as a set of distinctive features all arranged in a sequence.

So it's a matrix of features.

Going down in the columns we have our distinctive features.

And going across we have time.

So as the first thing Sussman and Yip did in their effort to understand how phonological rules could be learned is to design a machine that would interpret words and sounds and things that you see so as to produce the sounds of the language.

So they imagined the following kind of machine.

The machine has some kind of mystery apparatus over here that looks out into the world and sees what's there.

So I'm looking out in the world and I see two apples.

So what this machine might do then is, at some point, decide that there are two apples out there.

Then, thinking in terms of these guys as computer engineers, they think in terms of a set of registers that hold values for concepts like noun and verb and plural.

And we've not done anything with the machine yet.

We've provided no input.

So those registers are all empty.

Then, up in here, we have a set of words.

And they're all kinds of words.

Apple is one of them.

And those words up there know about how the concept is rendered as a sequence of a phones, that is to say a sequence of distinct features.

Then, over here, most importantly, they have a set of constraints.

So we'll talk about a particular constrain, the plural constraint.

Plural constraint number one.

And it's going to reach around and connect itself to some other parts of the machine.

Finally, there's a buffer of phones to be uttered.

And they're going to flow out this way to the speaker's mouth and get translated into a acoustic wave form.

So those are the elements of the machine.

Now how are the elements connected together?

Well, the words are connected, of course, into the buffer that is used to generate the sound over here on the far left.

The plural register is connected to what you see in the world.

What you see in the world is connected not only to plural register, but to all of the objects in the word repertoire.

This plural constraint here deserves extra attention because it's going to be desirous of actuating itself in the event but the thing observed in the world is plural.

There are lots of them.

So it's going to be connected then to the plural port.

There's going to be a z sound port down here connecting to that file element in the buffer.

And finally, over here is going to be a plussed voiced port, which is going to be connected to the second phoneme in the sequence.

That's how the machine is going to be arranged.

An of course, this is just one of many constraints.

But it's a constraint that has a very peculiar property.

Information can flow through it in multiple ways.

So we think of most programs as having an input and an output.

But I try to be careful to draw circles here instead of arrows.

Because these are ports and information can flow in any direction along them.

What I want to do now is to show you how this machine would react if I suddenly present it with a pair of apples like so.

So the assumption is that the vision apparatus comes in and produces the notion, the concept, of two apples.

So once that has happened-- that's operation number one-- then information flows from that meaning register up here to the apple word.

So that's part of stage number two.

Another part of stage number two is information flows along this wire and marks that as plus plural.

So operation number one is the activity of the vision system.

Activity number two is the flow of information from that vision system into the word lexicon and into this plural register.

So far so good.

Here's activity number three.

This word is also connected to the registers.

And information flows along those wires so as to indicate that it's a noun but not a verb.

That's part of part number three.

At the same time, part number three, information flows down this wire and writes a-p-l into those are elements of the buffer.

Now this constraint up here, this box, says, well, I can now see some stuff in that buffer that wasn't there before.

So it says, do I see enough stuff on my ports to get excited about expressing values on other ports?

Well, let's see.

What has it got?

It's got the elements in this buffer.

Also up here in step three flow the plural thing.

So it know that the word is plural.

So it says, is this voiced?

P is pa.

That's not voiced.

Is this a z sound.

No, that's not as z sound.

So it sees what it likes on only one of its three ports.

So it says, I'm not going to do anything.

I'm [INAUDIBLE].

I'm not in this particular combat.

So far so good.

What happens next?

What happens next is that some time passes.

And the elements of the buffer flow to the left toward the speaker's mouth.

So we get an a, p, l.

Same as we had before, but shifted over.

Now what happens?

Now what happens is that the l is now in the penultimate position.

So information flows up here.

Item number four-- oh, I guess that's item number five.

Item number four is the leftward flow of the word.

So in phase number five, the p is witnessed by this constraint.

p is-- sorry, l is witnessed by this constraint.

We moved it over one.

L is lll.

L is voiced.

So we have some flow up here like that.

That's number five.

Now we have voiced and we have plural.

And we have nothing here.

So there's a great desire of this buffer to have something written into it.

So now there's a flow down in there, of z, as item number six.

So that's how the machine would work in expressing the idea that there are apples in the field of view.

Mmm.

Real apples.

Not plastic imitations.

So that's how the machine works.

But all those connections are reversible.

So if I hear apples then I get the machine running backwards and my visual apparatus can imagine that there are apples out there.

That's how it works.

That's just by way of background the machine that they could see it for using the phonological rules once they're learned.

All the phonological rules are expressed in these constraints.

But since these constraints are such that information can flow in any direction, they deserve to be called propagators.

And in the good old days when everyone took 6.001, they learned about propagators as a kind of architecture for building complex systems.

But in any event, there's the Sussman-Yip machine.

And now comes the big question.

How do you learn rule rules like that?

Well, what we need is we need some positive examples and some negative examples.

And for the simple classroom example I've chosen the same challenge that I presented to Krishna.

We're gonna have cats and dogs.

So we're gonna look at the distinctive features that are associated with those words.

Syllabic.

Voiced.

Continuent.

And strident.

Just four of the 14 features that are associated with each of the sounds on those words.

Could you close the laptop, please?

Just for the distinctive features that are arrayed in those words by way of illustration.

So here we have k-a-t-z.

Phonetically spelled.

And if we work that out, let's see.

What is syllabic?

That's not.

That is.

That is.

That's not.

Voiced?

Ka.

Nope.

Ah.

Yep.

T. Nope.

Z. Yes.

That can't be right.

Cats.

I misspelled it.

Because cats.

Sss.

His a hissing sound but there's no voicing.

So that's not as z sound.

That's an s sound.

So that's not plus voiced.

It's minused voiced.

Continuent.

Let's see.

Is my mouth open when I say k?

No.

Ah?

Yes.

T?

No.

S?

Yes.

And strident.

Minus, minus, minus, plus.

It's only with the s sound that I have that kind of jet forming with my tongue.

Now we can look at dogs.

And now we have the z sound as the pluralization.

We know that because when we say it, dogzz.

Yep.

There it comes out as a-- we're only gonna look at the last two columns because they're the only ones that are going to matter to us.

So that's plus.

And that's minus.

Gu, gu, gu, gu.

That's plussed.

And that's plussed.

They're both voiced.

Is that right?

Dogu?

Gu.

Gu.

Is g sound voiced?

Yeah, I didn't think so.

G sound is voiced?

Look-- oh.

Oh, it is voiced buy it's not a continuent.

Just like that.

Yeah.

Cat, dogu zz.

Yeah.

It is voiced.

And it has to be for my example to work out.

And that's minus, minus, minus, plus.

So what we're interested in is, how come one word gets an s sound and how come the other words gets a z sound?

Well, it's a pretty sparse space out there.

We've already decided that there are 14,000 possible phonemes and there are only 40 in the language.

So that's one thing we can consider.

The other thing that we can think is that, well, maybe this is a logical problem.

Like the kind of problem you'd face if you were designing a computer.

And so Sussman and Yip got stuck for three months thinking about the problem that way.

Couldn't make any progress whatsoever.

And that happens a lot when you're doing a search.

You think you've got a way of approaching it.

Try to make it work.

You stay up all night.

Stay up all night again.

Still can't make it work.

Eventually, you abandon ship and try something else.

So then they began to say, well, let's see.

All we care about is the stuff before the two ending sounds.

We care about that part of the matrix.

And we care about that part of the matrix.

And we can ask, in what ways are those things different?

And they're different all over the place.

That's why they're different words.

We can ask the question a little bit differently.

And we can say, what can we not care about?

And still retain enough of an understanding of how the words are different so as to put the proper plural ending on them.

And they worried about that for a long time.

Couldn't find a solution.

The search space was too big.

And then they said, maybe what we ought to do is we ought to think about generalizing this guy here so that we don't care about it.

So now we don't care about that guy.

And then he went down through here saying, well, let's see when we have to stop generalizing.

Because we've screwed everything up and we can no longer keep the z sound words separated from the s sound words.

So that eventually distilled itself down to the following algorithm.

First thing they did was to collect positive and negative examples.

And there's a positive example and a negative example.

That's not enough to do it right.

But that's enough to illustrate the idea.

So the next thing they did was something that's extremely common in learning anything.

And that is to pick a positive example to start from.

It's actually not a bad idea in learning anything to start with a positive example.

So they picked a positive example and they called that a seed.

So in our particular case, cats is going to be our seed.

And the question we're going to ask is, what are the words that get pluralized like cat?

So we've got a positive and negative example.

We've picked a seed.

And now, the next step is to generalize.

And what I mean by generalize is you pick some places in the phoneme matrix that you just don't care about.

So you may pick a positive example.

And you don't care about it.

So you change it to an asterisk or, as demonstrated in the program I'm about show you, a ball.

Or you pick one that's negative and you turn it to a ball.

Bo.

So cats, this seed, becomes a pattern.

And in order to pluralize the word this way, you have to match all the stuff in here.

But now what we're going to do is we're going to gradually turn some of those elements into don't care symbols until we get to a point where we've not cared about so much stuff that we think that we pluralize that one with an s sound too.

So we keep generalizing until we cover, that is to say we admit or match, a negative example.

So that's how it works.

So we generalize like crazy.

And as soon as we cover a negative example, we quit.

Otherwise, we just go back up here and generalize some more.

And now we've got to pick a search technique to decide which of these guys to actually generalize when.

We could pick one at random.

And they tried that.

It didn't work.

So what they decided is that the thing that influences the pluralization most is the adjacent phoneme.

And if that isn't the thing that solves the problem, it'll be the one next to that.

So in other words, the closer you are, the more likely you are to determine the outcome.

So these guys over here are least likely to matter.

And those are the ones that are generalized first.

So if we do that, what happens?

Looks like we're going to come in here and see that there's a big difference between the non-voiced t and the voiced g.

But that's only a guess because I've only shown you a fraction of the 14 distinctive features that are involved.

So I suppose you like to see a demonstration.

Yeah.

So there's our 14 features.

And that's our seed there, sitting prominently in the display with pluses and minuses indicating the values of the distinctive features for all three of the phones involved.

That funny left bracket isn't a mistake.

That's just one convention for rendering the ah sound in cat.

So it's pretty hard to tell from just that matrix what's going to be the determining feature that separates the positive examples from the negative examples.

You notice that there are actually two examples down here.

There's cat and duck.

Is ducks got an s sound?

Ducks?

Yep.

So dogs and ducks.

They both get pluralized with an s sound.

And then we have beach doesn't.

That's beaches.

Dog.

We know that's a z.

Gun.

Gunz.

So that's not in the group.

So we can run this experiment.

Now here we go.

We're generalizing like crazy.

Generalizing, generalizing, generalizing from left to right.

So nothing in the first two columns matters.

Now we get to the t.

Wow.

There it is.

So it looks like you pluralize with a s sound.

The sss.

If, and only if, you're not voiced and you're not strident in the second to the last-- in the last phone of the word that you're trying to pluralize.

So that's one phonological rule that the system has learned.

And guess what?

It's the same rule that's found in phonological textbooks.

So now we can try another experiment.

So this time we're trying to deal with dog and gun.

And our negatives are what was previously positive plus beach, which is still in there as a negative example.

So let's see how that one works.

Nothing matters except for the last column, the last phone.

And now we find out that if the last sound is voiced, then the pluralization gets the z sound, a voiced determinator.

And finally, just to deal with beaches.

That's beach in it's funny phonetic spelling.

So now, if the final sound in the word is strident, if its got this jetty sound-- beach.

Beach.

Then it gets the ea sound.

So let's go back to experiment number one.

Because I want to point out one small thing about the way this works.

You'll notice that it talks about coverage and excluded down here in the lower left-hand corner.

Excluded, well, there are three negative examples, so they better all be excluded.

You don't want to cover any of the negatives.

But it says coverage, two and two.

That's because it actually is doing-- and now we have the vocabulary to say it quickly-- it's doing a beam search through this space.

So it's not just doing a depth first search.

It's doing a beam search so as to reduce the possibility of overlooking a solution.

So it says, oh, the coverage.

Both of the beam search elements cover both of the positive examples.

And they, in fact, have converged to the same solution.

So that's how the Sussman and Yip thing worked.

And then the next question to ask is, of course, why did it work?

And so the answer, as articulated by Sussman and Yip-- or rather more by Sussman.

Or rather more by Yip and a little bit less by Sussman.

Yip thinks that it worked because it's a sparse space.

And when you have a high dimensional sparse space, it's easy to put a hyperplane into the space to separate one set of examples for another set of examples.

So let's consider the following situation.

Suppose we have a one-dimensional situation.

And we have two white examples and we have two purple examples.

Well, too bad for us you can't separate them.

Now suppose that this is actually the projection of a two-dimensional space that looks like this.

Here are the white examples down here.

And here are the purple examples up here.

Now it's easy to see that you can separate them with just a line that goes across like that.

Now let's take this one more step and suppose that this is actually a projection of a three-dimensional space.

It looks like this.

This will be dimension one.

This'll be two going back there.

And this will be three up here.

And suppose that the positive examples are right here on this line.

Let's say this is-- well, we're gonna draw a little old cube like so.

Those are purple examples that are up there.

How many ways are there of partitioning the space along those axes?

Well, now they're not even just two.

They're three.

So one way to separate the purple from the white is to draw a hyperplane-- or in this case it's a three dimension, so a plane-- through here on the number three axis.

You could also put a plane in on that axis.

Or you could do both.

So in one case your dividing line would be-- let's see.

On the first axis that would be 1/2.

And then the don't care.

Don't care.

Another solution that would be don't care.

And then we divide on the number 2 axis with a plane at 1/2 and don't care.

Or we could do it with 1/2, 1/2, and don't care.

So the higher the dimension of the space, the easier it is sometimes to put in a plane that separates the data.

That's why Sussman and Yip think that we use so little of possible phoneme space.

Because it makes the thing learnable.

That's one possibility.

So one explanation for sparse space is learnability.

There's another interesting possibility, and that is that if you have a sparse space, high dimensional space with 14 dimensions, and if the 40 points of your language are spread evenly throughout that space-- now let me say it the other way.

If they are placed at random in that space, then according to the central limit theorem, then they'll be about equally distant from each other.

So it ensures that the phonemes are easily separated when you speak.

But if you go to ask a linguist if that's true, they don't know.

Because they're not looking at it from a computational point of view.

Well, we can look at it from a computational point of view.

So I did that.

After Sussman and Yip published their paper.

And here's the result.

This is a diagram that shows all of the phonemes that are separated by exactly one distinctive feature.

So if you look over in this corner here, you'll see that the constants-- w and x-- are separated by exactly one distinctive feature.

So they're not exactly distant from each other in the space.

On the other hand, they are pretty easy to separate relative to the vowels.

Which are here in this part of the diagram.

Which are all tangled up and the vowels are all close to each other.

So guess what?

Vowels are much harder to separate than constants.

Not surprisingly, because there are many pairs of them that are different.

And only one distinctive feature.

All right.

So now you back up and you say, well, gosh.

That's all been sort of interesting.

But what does it teach us about how to do science and stuff?

And what it teaches us is-- this is an example.

Ow.

This is an example which we can use to illuminate some of thoughts of David Marr, who I spoke of in a previous lecture, connection with vision.

But here's Marr's catechism.

I can't spell very well so I won't try to respell it.

But this is Marr's catechism.

So what Marr said is, when you're dealing with an AI problem, first thing to do is to specify the problem.

Gee, that sounds awfully normal.

The next thing is to devise a representation suited to the problem.

The third thing to do, vocabulary varies, but it's something like determine an approach.

Sometimes thought of as a method.

And then four, pick a mechanism or devise an algorithm.

And, finally, five, experiment.

And of course, it never goes linearly like that.

You start with the problem and then you go through a lot of loops up here.

Sometimes even changing the problem.

But that's just the scientific method, right?

You start with the problem and you end up with the experiment.

But that's not what people in AI, over the bulk of its existence, have tended to do.

What they tended to do is to fall in love with particular mechanisms.

And then they attempt to apply those mechanisms to every problem.

So you might say, well, gee, neural nets are so cool.

I think all of human intelligence can be explained with a suitable neural net.

That's not the right way to do it.

Because that's mechanism envy.

You fall in love with mechanism.

You try to apply it where it isn't the right thing.

This is example starting with the problem and bringing to the problem the right representations, gosh, distinctive features.

Once we've got the right representation, then the constraints emerge, which enable us to devise an approach, write an algorithm, and do an experiment.

As they did.

So this Sussman-Yip thing is an example of doing AI stuff in a way that's congruent with the Marr's catechism.

Which I highly recommend.

They could have come in here and said, well, we're devotees of the idea of neural nets.

Let's see if we can make a machine that will properly pluralize words using a neural net.

That's a loser.

Because it doesn't match the problem to the mechanism.

It tries to force fit the mechanism into some Procrustean bed where it doesn't actually work very well.

So what this leaves open, of course, is the question of, well, what is a good representation?

And here's the other half Marr's catechism.

Characteristic number one is that it makes the right things explicit.

So in this particular case, it makes distinctive features explicit.

Another thing that Marr was noted for was stereo vision.

So in that particular world, discontinuities in the image, when you go across an edge with the things that were made explicit.

Once you've got to a representation that makes the right things explicit, you can say, does it also expose constraint?

And if you have a representation that exposes constraint, then you're off and running.

Because it's constraint that you need in order to do the processing that leads to a solution.

So don't have the right representation.

If it doesn't expose constraints, you're not going to be able to make a very good model out of it.

And finally, there's a kind of localness criteria.

If you have a representation in which you can see the right answer by looking at descriptions through soda straw, that's probably a better representation than one that's all spread out.

It's true with programs, right?

If you can see how they work by looking through a soda straw, you're in much better situation to understand something if you have to look here and there and on the next page and in the next file.

So all this is basically common sense.

But this is kind of common sense that makes you smarter as an engineer and scientist.

Especially as a scientist because if you go into a problem with mechanism envy, you're apt to study mechanisms in a naive way and never reach a solution that will be satisfactory.