Guest lecture: James Wu

[00:00:00.79] My name is James Wu. I'm part of the UDub bioengineering program. I also work with professors in computer science, in BioE and in applied math, as well.

[00:00:15.29] I wanted to present this concept for the class, which is that a CPU is just a rock we tricked into thinking. Now this is a humorous quote, of course. But it conveys a lot of truth and a lot of commonalities that we must think about in your engineering between what we consider thinking and what computation is and the nature of computation itself.

[00:00:39.79] We might not think of computers as very similar to human beings in terms of thinking. But that's probably because we do not think about the nature of thinking very deeply.

[00:00:53.21] So what is thinking? We have to consider what we mean when we do this. We have some senses that we capture from the real world, either by touch, hearing, vision, etc. Then we consider some sort of goal that we need to reach. And then we plan out and execute a set of motor movements.

[00:01:18.49] Now I say motor movements, even though your actions might be muscular. It might be in terms of speaking. Those are also motor movements of your mouth and vocal chords. And you might also be doing things like gesturing or communicating via another device. But all of these actions are things that any computing device must also undergo, the sensing, processing, and planning the execution loops.

[00:01:56.15] So let's look at some examples of what thinking might be.

[00:02:04.77] These are three appliances, a light bulb, a heating element, stove. You might not think of them as necessarily thinking, but in actuality, they may contain elements or, in fact, all of what might be requisite of what we call thinking. For example a space heater, shown there on the left, you may not have come into contact very much, if you don't live in a cold climate. But it has a sensor for sensing the temperature of the room. It has a very simple digital logic. Either the temperature drops below a certain threshold and it turns on, or the temperature rises above a certain threshold that it internally knows about. And the heater turns off. And finally, has an execution, has something that it executes, which is to turn on the heating element, thus making the room warm.

[00:03:07.56] This sensing, processing, and execution loop, as you can see, is in all of our even simplest appliances. Now a light switch might only have part of that group in that it has an output. It has some sort of logic that you might control. And it does not have a sensor. You would be the sensing and control elements. But it will execute. Similarly, ovens do the same thing as the heater we mentioned before.

[00:03:39.87] We want to it focus, in particular, about the middle part, the processing, because while heaters and ovens have very simple processing in terms of sensing the state and converting that into an output, in which case, this is just on or off, other devices that were more used to in our daily lives have a lot more processing. And in a sense, what we're doing is using power,

using some sort of amount of driving gradient, the electricity in our walls, some converted form of solar energy, or some other type of source of energy in order to drive computation downhill, as it were. Like water flowing downhill, we drive electricity through a carefully planned set of routes in order for it to do decisive actions. And we call that computation.

[00:04:43.62] So here are some examples of what we used modern electrical infrastructure for. The driving force comes in from the wall and goes through billions of well planned routes. We call them circuits. And that converts into all of the actions that we represent as either apps or even sound over the audio waves or visual images over the TV.

[00:05:15.12] But in essence, this is no different from the simple flipping logic that was in the heating element. So how did this happen? How did the simple logic that are present in our everyday appliances turn into these marvelous examples of complex computation?

[00:05:33.96] Well, in the next segment, I'm going to tell you how brains appliances, dominoes, and even iPhones are similar. And that is, what we do is we take a basic computing pattern, the basic elements of logic itself, and place them in complex structures inside a device, such that computing patterns begin to emerge. What are some examples of this?

[00:06:02.28] Well, those of you who game may be very familiar with this type of example. So on the bottom, I want you to pay attention to is an example of a digital CPU implemented in Minecraft. This is kind of a really fun and almost ridiculous example, because Minecraft itself is, of course, a game that runs in a computer. But for those of you who don't know, Minecraft allows you to play virtual blocks and build things at will in this virtual universe.

[00:06:37.23] One of the things that allows you to place down is a mysterious substance called Redstone. And Redstone, a little bit, has the properties that we attribute to electricity. It doesn't flow as quickly. So that in game, you can still manipulate it and understand what's going on. But essentially, it's almost like electricity flowing down a wire. And using a combination of Redstones, you can actually form logical gates, such as AND gates or OR gates that form the basis of what we think of as computing.

[00:07:16.55] So using those basic structures which are in one of these diagrams players have helpfully compiled, you can build really, really large contraptions. In game, this would be hundreds of meters long with respect to your virtual character. But as you place down these blocks in careful order, you can actually make calculators. You can make contraptions just like the appliances we talked about earlier. That will be a lot simpler. All the way down to a complex CPU, what we call a processing unit.

[00:07:54.17] So this might seem daunting if you don't have experience with making digital logic into CPUs, but a very simple example that a lot of people can intuitively understand is above. So on the top, what I'm showing are a YouTube video of a YouTuber making logic circuits out of dominoes.

[00:08:20.45] So dominoes are something that we do understand. You knock one over. It triggers the next one over and so on. But if you place them in careful patterns, you can also make logic

appear. How? Well, on the right hand side, you see a diagram that says A or B. That is a basic word logic gate. What does that mean?

[00:08:44.16] Well, whether or not. So if you imagine it as being tripped from the bottom up to the top, whether or not you trip the left hand route or the right hand route, you will always end up with collapsed dominoes in the middle top. So if you consider the left hand of the route as A and the right hand as the B, and the top as the output, A or B will always result in C being tripped. Now this is very simple, of course.

[00:09:18.20] But that's just one simple building foundation block. And if you weighed it out in very complicated structures, like you see in this photo to the left, which you can check out on YouTube, you can create even more complex mechanisms. On the right is a XOR circuit, so it will trip if the left hand side or the right hand side, from the bottom, is tripped. But not if it's tripped both at the same time. You can imagine in your head a little bit how this works. If you trip it from the left hand side, it will flow through to the top. Same thing if you do it from the right hand side. But if you trip both at the same time, these two pathways of dominoes will actually jam up in the middle and not trip up the top.

[00:10:08.36] Why are these important? Because if you place enough of them in a careful enough manner, you can get complicated things like addition, subtraction, multiplication, and so on. So what if you wanted to do this faster? Of course, dominoes are super slow. And actually so are Redstone inside this game. It kind of has to be so that you can keep track of what's going on.

[00:10:31.33] Well, what we've also made are things like accelerators. Now these are more historical objects. But once upon a time, not very long ago, less than a generation ago from this lecture, we had digital computers. We did not have digital computers, rather. And we had that mechanical computing. And people would drive these mechanical computers.

[00:11:00.40] So on the top is an example of a slide rule. If you've never seen one before, you might see it in older movies, like Apollo 13, where it's prominently featured. But what people have discovered is that you can do multiplication and division very quickly, without doing it longhand on a piece of paper, just by putting marks on a ruler, logarithmically.

[00:11:25.96] Why is this important? Well, you know that if you take two rulers, you can actually do addition and subtraction, because if you have three inches on a ruler and three inches on the other ruler, that gets you 6, if you line them up side by side. You can do the same thing if you added algorithmic, if you draw the marks not linearly, like 1, 2, 3, 4, but logrithmically, 1, 2, 3, 4 with spacing decreasing as the numbers get higher.

[00:11:58.78] And this is a very clever way, because if you line up the marks, 1 here with the 3 here, it will be like multiplying by 3 on the top. And I invite you to try any online slide rule or auto slide rule apps, which you can download for your phone and try out how the length add up to do multiplication and division.

[00:12:25.71] On the bottom is another example of an accelerator that we built into mechanical space. This is a Curta. It's an early computer. It was actually a very expensive computer at the

time, or more appropriately a calculator that can do addition, subtraction, division, and multiplication simply by turning gears. So there's a lot of stuff on this diagram. And I invite you to check out a YouTube video of how this works.

[00:12:57.62] But essentially, you set different dials and gears and turn the handle. And it will do the basic arithmetic operations for you inside, simply by using gears, which we know can multiply and add and divide.

[00:13:16.33] To the left is an example, is a simpler example, of a marble computer, which if you YouTube for a marble computer, you can also find. And essentially, this is an example of digital computation, digital binary computation. Here, digital doesn't mean electronic. Digital just means on or off states, much like the heater circuit we were talking about, except that heater circuit has one bit.

[00:13:45.13] In this marble example, there are three levers shown. The total contraption actually had four. Three levers shown. Each of these is a bit. And you can actually set them independently, using marbles. And thereby forming a binary adder that is able to compute up to 2 to the 4th power, because of how binary logic works.

[00:14:13.87] So if we can make all of these things appear in mechanical space. And in the example of the Curta, or in the example of the slide rule, do simple computations. I call them simple, because you can do them longhand on the sheet of paper. It's just very slow. And we can do the simple computations really, really quickly, faster than any human can do them. That means what we made is an accelerator, something that takes a simple set of operations and do them much faster than we can imagine. And thereby producing almost magical effects of getting from the sensor to the output almost immediately.

[00:14:59.92] So do humans have accelerators? Do brains have accelerators? Because we've just shown that we can do computing patterns in Minecraft, in marbles, in dominoes, and even in length of a ruler, if you place the marks carefully. So it must mean that human minds also contain these type of underlying computing patterns.

[00:15:27.49] Can we look for examples of them, things that we do faster than we can possibly reason about them, something that almost happens immediately to us.

[00:15:38.26] I want you to think of a few examples on your own. But one of the key things, one of the prime examples of this, is facial recognition.

[00:15:48.91] So why is this so important in computing as a basic example of an accelerator? Well, like I've shown before, where a slide rule or that Curta gear addition, subtraction machine doesn't really do anything unimaginable. All it does is addition and subtraction and really, really quickly. But the output is faster than any human can ever reason out. The same thing really happens with facial recognition.

[00:16:23.87] You can, afterwards, really really excruciatingly slowly describe what makes a person look like themselves, the eyes, the spacing, the mouth, the hairline, etc. the facial bone

structure. And you might be able to, over a period of hours-- I mean, we call that doing police sketches-- describe what makes a face look like someone or doesn't look like someone.

[00:16:51.92] The action of recognizing someone really happens to you. It's so fast you don't really even realize what just, what your brain just actually did. And this is something that we've been trying to replicate in computing for a long time. And we've only been able to recently do them with the advent of deep learning with any sort of speed and accuracy.

[00:17:16.23] This is an example of an accelerator at work, a series of computations that, with dedicated hardware in portions of her brain, like the fusiform gyrus, that run so fast that we cannot really describe what is going on deep down, until we take vastly more amount of time.

[00:17:40.26] This is an example that is in neurological hardware, our brains, as well as even in our consumer hardware. So as the Curta or the slide rule was to arithmetic, today we have dedicated chips and dedicated cards for different types of computation. You might have heard of a graphical processing unit or a GPU. Or sometimes, you might just hear it as a video card, if you're a gamer.

[00:18:12.89] Those are hardware that are specifically built for the purpose of computing things like light, 3D geometries, and things like visual output much faster than your CPU can do them. In this case, you can kind of think of the CPU as describing how to recognize the face, whereas a graphics card might be just doing it extremely quickly and handing you the end result.

[00:18:45.31] Your brain does the same thing. And so what we might conclude with the brain, or learn from the brain, is how exactly we are able to use all of these accelerators that might be present in our brain. How do we know there are accelerators? Because we can do all of these actions much faster than we can ever reason about them. And how we might be able to reverse engineer or try to figure out how these accelerators in our brains work, in order to be able to do some of the similar marvelous tasks in computing for people who might be paralyzed, who might have injury, who might have neurological disease. And this is part of the goal of the Center for Sensory Motor and Neuroengineering.

[00:19:42.52] The problem is that the architecture of brains is very different from the architecture of CPUs in Silicon. In general, when we program or design hardware in Silicon, we use a bit more of a sequential architecture. This is a little bit simplifying the architecture of what happens in computing. But in comparison to neurons and biological structure, we have a very simple sequential and quick architecture in comparison to a relatively slow, but tremendously parallel architecture.

[00:20:22.81] What do I mean by that? Well, when we program a computer to do is a set of serious and complex tasks, generally, we have a set of groups or a set of linear pathways that runs really, really quickly.

[00:20:44.23] Your phone might do this. For example, it might wait around and in a group, keep checking for an action, for example, user interaction, you touching the touch screen, clicking a

mouse, typing on a keyboard. Once it senses that, then it carries over those discrete actions over to some other process.

[00:21:05.18] The architecture of brains cannot afford to operate in this way. Why not? Because neurons are slow. They don't use electrical activity directly. Instead, we use electrochemical reactions that, instead of running at the speed of light or propagating, transmitting at the speed of light, our neurons in our bodies and our brains operate much more close to the speed of walking or running.

[00:21:37.89] This is dependent on whether or not the neurons myelinated. But it's safe to say that, on average, in your brain, electrical signals or electrochemical signals are propagating roughly around the pace of walking. So to overcome these limitations that are a result of biological evolution and still do extremely complex computations, like facial recognition or motor movement, which I'll go into in a second, our brains have evolved tremendously parallel architecture. It operates and does computations all at once and wait for them to finish all at once, in order to result in the actions and the planning and execution that we use to do our everyday lives.

[00:22:29.53] This is a very important distinction, because no matter how parallel we make our silicon architecture, we have not remotely come close to a type of parallelism that exists in brains. And this is a very sticky point in terms of being able to develop technologies that perform neurological behaviors as quickly, as well as being able to read out from the brain.

[00:22:53.82] We [? at the CSNE, ?] we build or try to lay down the foundations of neural implants. But while it is relatively easier to tap into a piece of electronic, because we know that at a certain point, we might-- it might be conveying something we can understand, such as if we tapped into this loop here, we might be able to detect user events, like you touching the touch screen. But if we tapped into any individual neuron or even clusters of 200 300 neurons, you might be only getting a very, very small part of the picture, because the rest of the activity is happening elsewhere in this tremendously parallel architecture.

[00:23:41.12] So let's combine what we have learned so far. What we learned is that the neural architecture is a lot more difficult to read from than a silicon base or a digital architecture that we use in modern electronics, because we need to look at more of the brain, vastly more portions of the brain, to get the whole picture of what the brain is doing at any period of time, because it is a lot more parallel.

[00:24:12.23] In addition, much like modern computing our brains also use accelerators or computing motifs or competing patterns that perform tasks way faster than we can reason about them. The combination of these two means a couple of implications, all of which we run into when we try to engineer neural implants for brains.

[00:24:40.68] So this is an example. There is a huge performance difference between modern computing and its ability to do what we consider simple motor tasks and what humans are able to do.

[00:24:54.69] So on the left hand side, you seen an animated image of a robot trying to clean a dish. In fact, this image is actually sped up. On the right hand side, you see a human girl, albeit a world champion in cup stacking. But you can see the tremendous amount of speed difference in being able to manipulate common everyday objects.

[00:25:20.25] The left robot took a PhD project to complete. In fact, it took many PhD projects at a world class robotic institution to complete. Why did it take that much effort?

[00:25:32.91] Well, the everyday tasks of taking two objects of different shapes, such as a cup and the sponge, in this case, which are not rigid, but compliant, in the case of the sponge and have it sense, using a variety of sensors, amount of pressure, torque, and force that are transmitted through the sponge handle, is a mind-bogglingly large amount of competition. And that's why it took so much effort to even get a robot to slowly wash dishes.

[00:26:09.66] On the other hand, our brains have accelerators, have motor accelerators, that are purpose built to sense things like pressure and shear and force and compliance and allow us to do things like cleaning dishes very, very quickly and very effectively. But if you describe to your computer an algorithm for turning all of these into forces that won't break the glass or won't clean the glass incompletely, you would have a very, very hard time.

[00:26:40.63] This brings us to an early realization in the 1980s by a famous roboticist at Carnegie Mellon, Hans Moravec, who coined the Moravec's theorem or the Moravec's paradox, that it's comparatively easy to make computers exhibit adult level performance intelligence test or playing checkers and really difficult or impossible to give them basic skills of even a one-year-old.

[00:27:12.17] This is known, well-known, back in the 90s. And it still is true today. Things that we think of as very simple, such as movement and washing dishes or cooking, are actually really, really difficult computational problems that we simply had and evolved accelerators over millions of years in order to perform without us thinking about it. And now that we have to reason about them, in terms of, for example, building devices to rehabilitate patients, whose brains have been damaged or whose limbs have been damaged in some way, then immediately we realize how difficult this problem is, when healthy individuals don't really even have to think about how to perform these tasks.

[00:28:08.55] And why does this matter? Well, amputation and paralysis are things that affect millions of patients. And the current standard of care are some very poorly controlled prosthetics, such as the hook hand, which really is operated by steel cable and the position by which you hold your elbows and even very simple myoelectric devices.

[00:28:34.93] So in the middle you see a photo of a myoelectric electric device. What it does is sense, very poorly, muscle activity, in general, so that you can open and close the hand, which does not have more than one degree of freedom. Basically, all it does is open or close. There is no individual finger movement, no adjusting, everything that you really need to do everyday tasks, like using a spoon or something like that.

[00:29:03.84] And for total paralysis or individuals with a much more profound quadriplegia, our standard of care is actually something like a sip and puff device, where, if the individual's facial muscles are not paralyzed, you sip and puff air from this tube in order to do a simple one degree of freedom control, like clicking a mouse. The rest is perform with things like eye trackers, presuming that the patient has some control over eye movement.

[00:29:38.04] So overall, the hard problem is not something like playing chess or doing abstract mathematics. Instead, it is doing things like cooking.

[00:29:51.85] So in this video, you can see a patient using the standard of care prosthetic, a splint hook, in order to tie their shoelaces. You can see how much practice this takes in order to accomplish something that we take for granted everyday, which is manipulation of a soft body object, like rope or string or shoelaces or even your headphone cables.

[00:30:18.44] My research focuses on the interaction of prosthetics and motor control. And this brings us to something we called, the control problem.

[00:30:30.32] The control problem is actually very well illustrated by a fun flash game that appeared about 5 to 10 years ago called, QWOP. It's called QWOP because it's a bit silly. You control a running person. But instead of controlling legs as your own legs, which you don't think about running very much when you're a healthy individual, you just sort of run forward.

[00:30:58.44] But if you had to control your legs individually, your muscles legs individually, and used something simple like Q and W, the keys Q and W to control your thighs, and O,M,P control your calves, then the problem becomes ridiculously hard, because it separates you from your motor accelerator. You no longer have the ability to do this action without thinking about it. Now you have to reason, at the very slow speed of reasoning. And most people who play this game end up as in this image, a very badly collapsed person who cannot run, because you are not able to think fast enough to control individual muscles with any degree of speed.

[00:31:53.75] Why is motor control difficult? Well, it's also difficult mathematically, too, because if you think about the problem of positioning two angles in order for an arm to reach a desired location, as you can see from the bottom part here, if you had [? two fixed ?] arm length, and all you were able to control are the angles of these arm length, and you wanted to hit a desired location, xy, it's a solvable, but a little bit more difficult problem.

[00:32:28.64] You might be able to solve this if you had taken trigonometry, for example. You would use two arctangents or arccosines to try to solve this problem. And this is a solvable problem with only two joints. But once it reaches three joints, it becomes a mathematically incompletely solved problem. That's because there are multiple solutions. And things like approximations become a lot more important in trying to design a robot that can solve for a three joint problem.

[00:33:08.27] Well, let's look at your hand. How many joints are present in the human hand? Well, that's a very, very large number of joints, actually, about 30 joints in one arm and hand

alone. And that you're probably able to actively control about 18 separate degrees of freedom. That is, 18 individual or axes of movement.

[00:33:37.09] You can try it out on your own hand or finger. In fact, even your thumb alone, you can bend it. You can turn it from side to side. You can also extend it out away from your palm. That is a lot of degrees of freedom just simply on the human thumb alone. And to solve the mathematical problem of positioning all of these joints in order to do common everyday tasks, like using a spoon or fork, is a very, very hard computational problem, without the use of the evolved accelerator that we all take for granted.

[00:34:16.05] So in order to restore function to patients who have [? lost ?] these types of upper limb motor function, we must first come close to understanding and solving control problems and understanding how that occurs in the brain.

[00:34:36.10] Now augmenting humans isn't new. From different museums, you can see showpieces of prosthetic toes, prosthetic noses, even, and very meticulously crafted arms that either might be mobile or not mobile, or sensory augmentation, like in an ear trumpet for people who have lost hearing.

[00:34:59.94] Now the modern design and intent to build things like cyborgs is a long, long standing one that extends throughout history. But like we discussed, in order to do any of these things, we must overcome all of these challenges. And furthermore, we must also consider their implications. And so just as we have to solve and understand and explore more of the neurological basis of massively parallel computation, that is, solving problems in everyday life in very, very many parts of the brain all at the same time, even though we currently only have the ability to read from one or a few spots at a time, like in this case.

[00:35:52.26] In the BrainGate program, where we've implied that some electrodes into just the one portion of what we think is the motor accelerator for the humans called, the primary motor cortex. To things like assistive exoskeletons that actually don't implant into the brain or the nervous system, but instead, just tries to sense your output and assist it, to things that are mounted, assistive exoskeleton on your arms that might respond to various different types of motions, as well as technologies that try to activate your muscles directly, via electrical stimulation or shock.

[00:36:36.42] All of these devices lie along an augmentation and an assistive axes. And some of her most successful technologies, in fact, are merely assistive and noninvasive. And what we are building towards is a future, as we understand more and more these spaces of computation and understanding the readouts from the brain, a more futuristic, invasive, and more augmentative set of tools that allow us to achieve possibly more than the current capacities of humans.

[00:37:24.60] Already, we have devices in our pockets, smartphones, that do already extend the capabilities of humans. It can access information and do computation faster than we can. But the traditional actions that we consider easy, that require accelerators, things like facial recognition, things like motor movement, we still cannot match with modern silicon. And I think that, one day, we will. But we really have to think about the ethical and sociological ramifications of

building these systems in the future, as well as working towards a more complete understanding of how neuroscience and technology might combine with each other, too, in the future.