SILICON MODELS OF NEURAL COMPUTATION

Carver Mead
California Institute of Technology

Thank you Steve. It's a great pleasure to be here. Every plenary talk, of course, has to open with a story. I'm reminded of the story of the Indian Chief who calls his braves together and says (this a good news, bad news joke), "Well, I have some good news and some bad news. Which do you want to hear first?" They decide they want to hear the bad news first. "Well," he says, "all we have to eat for the rest of the winter is buffalo chips. But the good news is that we've got lots of 'em."

I'm afraid this afternoon I'm in a little bit of a position of bringing some good news and some bad news about technology. I'll start with the good news. Thanks to microelectronics technology we have lots of computers at our disposal. And we have, as you can see from this conference, a lot of neural network programs and simulations and a lot of experiments that are running on those computers. That's the good news. There is some bad news that goes with it, as any of you are sitting around waiting for your latest simulation know, and that is that digital computers don't make very good neural networks. There are a number of reasons that. One of the real problems is discrete time. As all of you know, there are convergence problems with any non-linear dynamical system that's modeled in discrete time. Furthermore, there is no general solution to the numerical instabilities that result from the discreteness of the time steps. This is as distinguished from the stability of the dynamical system itself. So there's a big problem there. We all spend lots of computer cycles getting around that problem; making sure that our simulations are stable. There's another problem with the way computers work, and it is that the same properties that make computers really great at being general purpose can make them really abysmal at doing special purpose jobs. I'll come back to this as we go on.

In a real neural network, or any real dynamical system for that matter, time is its own representation. It isn't an index in an array somewhere, or a pointer to somewhere in memory. There are other problems. Even the parallel computers that are appearing on the market don't have the right grain size for a neural network. A synapse really does very, very simple computations, and its really a shame to tie up a whole processor which has memory management, floating point capability, and all that for for the operation of one synapse. There's another thing that isn't done very well, and

that is the precision or resolution. Neurons don't have very good precision at all, but they have pretty good dynamic range. If you're modeling things with floating point numbers, you can do that all right: you can have the unlimited precision and a large dynamic range. But not many computers in operation today are like that. Signal processing stuff is better.

The nervous system is very smart. It does a lot of things very well. It uses the physics of its devices -- its synapses, its voltage variable channels, its membranes, its salt water -- to advantage. Your first impression on looking at that icky, sloppy, gooey stuff in the nervous system is to say, "Yuck! That's awful stuff. Why would anybody build a system out of that?" But they work awfully well. And let me remind you that our icky, gooey, sloppy stuff with its millisecond time constants can outrun any super computer today on any but the simplest perceptual tasks. It uses time as its representation. That's a good start.

The processing in neurons is analog, and you get a lot for that. The continuous-time signals that you get that way don't alias away the information with any kind of discrete sampling. Many computations come free. For example, if its a compact neuron, you could model it as a single electrical node. And Kirchoff says that all currents that go into it get summed, so you get an add operation for free. The currents also get integrated with respect to time because of the capacitance of the membrane, so you get time integration for free. So there's a lot of nontrivial computation that you get free because of the laws of physics.

And then the devices themselves give you more stuff that is really non-trivial. Exponentials, hyperbolic tangents, and things like that come free because of the exponential nature of the Boltzmann distribution and the fact that all neural hardware with the sloppy stuff works with energy barriers through which the channels of the membrane have to go. The nervous system uses map representations in all the sensory modalities that we understand; it keeps things that are related to each other physically close. For that reason, it doesn't have to run analog wires over long distances. All the analog wires in the dendritic tree, where all the tough computation is done, are short. That has the great advantage that all the computations done in the nervous system are done in context. Any signal is always compared to the signals around it. I'll talk about that later. And oh, by the way, if you have to send a signal a long way, there is the white matter, the axons, where signals are turned into nerve pulses, sampled with respect to time, and sent over long distances. But you only do that when the representation is such that sampling won't hurt it.

After all this, it must be obvious to you that there's nothing about this that has the slightest thing to do with neurons, per se. It has to do with electrical signals in any physical environment. And in particular, we can do absolutely the same thing in silicon. By the way, suppose you never did know anything about bits, or Boolean algebra, or any of that. The beauty of bits is strictly in the eye of the beholder -- or the eye of the designer. All that effort we put into taking those beautiful analog signals and turning them into ones and zeros seems a little pointless after you look at the nervous system a while.

About 10 years ago, Marvin Minsky came to Caltech and gave us a wonderful, wonderful talk, one of the nicest talks I ever heard. And he started with a story, or a confession, depending on your point of view. He was speaking, of course, as one of the great pioneers of artificial intelligence. And he said, "You know, when we started artificial intelligence, we made a program that did a passable job of working college calculus problems. And so we thought we were on the verge of this great new era. So just to sort of fill in things a bit, we went back and decided to do high school algebra. And it was to our great surprise that we found it was harder. That should have been telling us something, but we went on to tackle grade school arithmetic. And we found, again to our great surprise,...(laughter). Even to this day, there are open research issues there, like the difference between numeral and number. There are a lot of issues that really aren't trivial. Then we tried the child's world of blocks, and we found that it absolutely defeated us unless we circumscribed the arena of effort very severely. And finally it dawned on us that most of what we call intelligence is acquired the first year of life."

That's a very important piece of insight and I'd like to draw the picture this way. There's a lot of talk at this conference about recognizing things. Now I'm all for recognizing things; I really think its a good idea. But recognition is a very, very cognitive idea; it's a very high level kind of thing. And we human beings naturally think in terms of cognitive ideas because we can perceive ourselves doing it. We can introspect and follow the process, and I guess I don't have to remind you that Allan Turing's original paper really said that he was building a model of a machine to prove theorems the way a mathematician proves theorems! So the cognitive idea of what all this stuff is about has been with us for a very long time. And it's a very wonderful goal. But let me remind you that cognitive processing, which is the little tip that sticks up above the surface of awareness, has underneath it an enormous iceberg of precognitive, preattentive processing.

This very parallel processing starts with the senses. Sensory inputs are all way down on the bottom of this iceberg -- way below the surface -- where it's real hard to see them. And it's real hard to see the gyrations they go through before they get to the cognitive end of things. I should remind you that this wouldn't have been the first time that we underestimated the size of the things that have to get done before we can get up to cognitive processing. That's another piece of bad news that I wanted to bring up to the conference. But just because we can't see ourselves doing cognitive sensory processing doesn't mean it isn't there, and there in massive amounts. It was the thing that defeated artificial intelligence in tackling all the really hard problems, and it can defeat neural networks if we don't take it on on its own terms. It's there, and it has to be done before we are going to be able to do the cognitive stuff. And it has to be done well. So let's talk a little about sensory input.

What really happens? Your genes don't have enough information in them to specify the way your brain's wired up. But they do have enough information in them to specify some of the basic structural entities in your brain -- the archetypes, if you like. These archetypes are the basic structures which will then develop, as sensory input passes through them, into the preprocessors of sensory input that feed the higher levels. The raw input goes up through this enormous iceberg -- many, many levels of representation -- before we ever do recognition. Most of those levels get wired up very early in life; like Marvin Minsky said, during the first year of life; earlier perhaps. The early visual system of a cat gets wired up in the first couple of weeks. (There are some wonderful stories about that.) The representation at a higher level depends on the development of a representation at a lower level. Obviously, this is so. Otherwise you don't have any inputs to drive development of higher levels. By the way, over and over again, time is an absolutely essential axis in the representation of all sensory modalities.

So it's probably becoming obvious at this point that sensory pathways develop from the sensors inward. When the animal is born, it is barraged by an incredible stream of sensory data. And that wires up the first level. And then that level wires up the next level, and gradually the whole thing comes together. It is no accident that evolution proceeded from simple structures to complex structures. Very much a bottom-up idea.

One piece of thinking that bothers me is the idea that, if we're so smart (after all, we have 2,000 extremely smart people here), why don't we just go off and do it top-down? You know, we'd probably do that, at least the first part of it, but the problem is that nobody knows where the top is!

There's a little bit known about representations in the nervous system at low levels. For early sensory processing stages, we know a little about the representation. We know less about the processing that leads to those levels. At high levels, there's a fair bit known about what the nervous systems does from psychology. But there's nothing known about how it does it. We known some vague, general things: it's generally thought that coordinate transformations go on in the parietal lobe. There are known to be some representations of shapes like faces and hands in the infant cortex. That kind of thing. But absolutely nothing is known about how they are used, a lot of the major pathways, and what representation they are carrying. No idea, except down near the sensory levels, where we know how to put inputs in that will cause something to happen. So we don't know where the top is. We have some idea about how the thing behaves as a whole. We have some ideas about what happens when you take away parts of it. But we have no idea how the computation is done, so we don't know where the top is.

So artificial intelligence got into a trap. And the trap is there because everyone, all of us, grossly underestimated the amount of processing between the senses and the cognitive levels. And I claim that if we're going to be successful in this endeavor, we're going to have to stay out of the AI trap. We had better drive our systems with real sensory input. It's really easy to make up an image and have a vision system recognize what you thought it ought to in the end. It's really hard when you expose that same vision system to real sensory data and see that it doesn't work anything like you expected. I've had that experience a lot so I can assure you.

So what's the AI trap? Well, the AI trap works something like this: You announce that you are going to do a really hard problem. And then you start working on it and you find out that, not only is it a very hard problem, it's ORDERS OF MAGNITUDE harder than you imagined. So then, of course, you do what every good scientist does: you go back to a problem that you can solve. When I went to school, a professor of mine said, "You keep simplifying a problem until it goes away, and then you go back one." Of course, you can solve that problem by definition, so you solve it. And then you make a little demo of that, which you can make pretty convincing nowadays, thanks to our beautiful digital computers. And then you announce to the press that you've solved the problem. And that would all be okay, except that you have to be careful never to reveal what the hard parts really were, because some of the hard parts don't look like they ought to be that hard, they really don't. And you feel really stupid, so you don't talk about that. And then you go on to step one of a more difficult problem.

Well I don't think you have to do that. I think that's the fastest way to ruin a budding field that has a tremendous amount of promise, that has the right kind of interdisciplinary interaction. This field has people that know a lot about a lot of things. I've gone around and listened to the talks at this meeting, and the level of expertise is extremely high. The diversity of background is extremely high. We don't need the AI trap; we really don't. But if we're not going to do it that way, we'll have to be really honest about getting real data. And you know, there is a lot of history in that.

This is a wonderful slide. Before artificial intelligence, people were doing stuff like this. This is a slide of a wonderful display that will be downstairs tomorrow. This was a retina built at RCA starting in 1961. It was a retina that was designed to emulate the operators that Jerry Lettvin proposed for the frog's eye in that wonderful paper in the IRE, back when radio was still an okay word. It is called, "What the Frog's Eye Tells the Frog's Brain". It is a wonderful paper, and is still good reading today. This retina had some seven to nine levels, depending on how many were working at the moment. It communicated between levels by light. It was four feet wide by four feet high. (In the slide,) someone's holding an image up in front of it, and here's what the retina saw in that image. There were simple versions of all the operations of a real retina implemented in the structure. This is a wonderful start. But there's a problem with it, and we can see something about the problem in the slide. It has to do with technology. This next slide is a little snapshot of what's happened to the technology as time has gone on. The retina that I showed you was four feet wide and four feet high because it was built with technology somewhere between there and there. It had photoconductive discrete devices, and neon tubes. It was a wonderful design. It had wonderful ideas. Very straight thinking, but the technology wasn't there.

But technology has moved a lot since the early sixties. This was what transistors looked like in the mid-to-late sixties. And this is a 1959 shot of the very first integrated circuit. What seemed to be a trivial invention, hooking up more than one transistor with the metal that was already there on the silicon, turned out to have an enormous set of consequences in technology today. We now have memories with a million bits on them. You can see them hooked up in the neurocomputers downstairs. We have microprocessors. This one has a half million transistors on it. And you can see some very ambitious microprocessors downstairs. This has changed our lives. It's changed the way we think about everything.

This is the learning curve that we went through with the technology. This learning curve, from bottom to top, grows a factor of about two a year in the number of transistors in commercial circuits you can buy. There's never been a learning curve like that in any kind of technology in human history. Oh, you get a curve like that for a rat population. But it's an awesome curve. It means that, by the time you turn around, there's 10 times more computing than you thought there would be. And we're out here somewhere, and it's slowed down: it's now only a factor of two every two years. And it's not going to stop for awhile.

This is an old picture out of a paper we did in 1971 where we took a cut at how small you can make a transistor. And it turns out that this is about a factor five smaller than they are today in linear dimension. That's a factor of 25 or so in the number per unit area. And wafers are getting bigger, so we've got maybe a factor of 100 more to go before we actually get to the limits imposed by the laws of physics on our technology. This prediction has been holding up pretty well, but I'm sure that once we get close, people will get smart and figure out ways to get around it.

What I'd like to show you is some pictures of an old piece of work -- this is three years old now -- that we've filmed to try to follow up the work that was done early in the sixties at RCA in building a physical retina. Except we want to use this wonderful new technology to do it. So what we really want is a little patch of retina. If we wanted the whole retina, we'd have to get a little bit further involved in the technology, and we'd have to use a whole wafer instead of just a chip. But that's okay, you can do that.

Let me just remind you what the retina looks like. It's got photoreceptors up on the top. The function of the photoreceptors is to transduce the light into an electrical signal. And on the way, they take the the logarithm of the intensity of the light. They take the logarithm and make that into an electrical signal. That's a wonderful idea. It gives you a wonderful invariant, namely that a given contrast ratio is represented by a given voltage difference. That's a wonderful thing, for it takes out the absolute illumination. Then the signals pass on down through these things called bipolar cells, down to the ganglion cell which is the line driver for the optic nerve. And on the way, it's intercepted by a couple of transverse layers that do very, very important kinds of processing. The first layer is called the horizontal layer -- horizontal cells -- and they do adaptation for the level of incoming signals. The second transverse layer, the amacrine cells, is a lot more complicated. There are some 20 or 30 different kinds of amacrine cells in higher animals, even in fish, and its not known well what they do. It's known

that some of them are responsible for taking the time derivative signals. They are also responsible for applying some gain control at that level of representation.

If you put a flashing light on the center of the retina, here is what you get. This slide is from a wonderful paper about 10 years ago in Scientific American by Frank Werdlan. It is still one of the nicest things written about the visual system. These curves show the logarithmic response. He plots log units of intensity along the horizontal axis and linear voltage on the vertical axis. You see a logarithmic response, with saturation top and bottom. This is just an illustration; I'll show you some real data later on. The surround is this whole structure here. It can be either bright or dim, and it can either be rotating or fixed. If you look at the signal of the bipolar cells, you find that the center of the response, the level around which deviations can be passed on to the next level of processing, is set by the surround. Whatever the intensity of the surround, the response curve centers itself on what it sees as the surround. This is the simplest kind of context processing. You'll find it in every step of the visual system until you get to extremely high levels, and probably there, too. Once we know it we'll know, but we don't now. Later on at the amacrine layer, what we find is that whether the surround is twirling around or not matters. We're taking the time derivative now of the signal, and if the time derivative of the surround is large, it will shut down the gain of the local time derivative circuit. In other words, it's an attempt to keep the gain of the system such that only the most important motion events are going to be recorded, and the others suppressed. A typical kind of lateral-inhibition gain control.

Now something about our silicon model of the retina. The first stage is the logarithmic detector. It turns out all this stuff is built in perfectly ordinary, garden-variety CMOS, the same process that is used for making processors, memories, and stuff. And in that process, just because they couldn't do it any other way, there's a parasitic bipolar transistor. That's really nice because its one of the world's best phototransistors. So built in to perfectly garden-variety CMOS processors is a wonderful photoreceptor. So we follow a lead from nature; we make a virtue of necessity. On them, we use the old electrical engineering trick of taking feedback from the output through a nonlinear element to get a saturating response. If that nonlinear element is exponential, then the response will be logarithmic. The exponential element we use in all the stuff we build is MOS transistors running sub-threshold. Their response, the current out of them, is exponential in gate voltage. They don't draw any input current. They're wonderful; they're just what we've always wanted. Something like this has about 320

mV per decade. And if you make such a photoreceptor, you get an exponential voltage output as a function of intensity. We see it goes over about four or five log units of intensity. You can see that the current level goes down below 10^{-14} amps. Those of you who are into electrical engineering things know that's a pretty good electrometer. The light intensity corresponding to that current is about where the cones in your eye give out. I can't compete with the rods; they'll count photons. But the cones we can get.

This is the basic structure of the retina. It has these receptors. Each of the receptors has, hooked up to it, a little local computation element. Those go through a horizontal resistor network that's modeled after the resistive network of a real retina. I'll show you the details now. What's recorded on the output, is the difference between the potential of the receptor and the potential of the network. I'll show you the circuit. This is the picture of a few pixels. You notice it's a hexagonal array. The little funny yellow square with the green "PAC" person in the middle of it is the photo receptor. There are a couple of amplifiers, a couple of horizontal resistors, and some stuff that takes the logarithm. So it's a very simple pixel. The chip looks like this. This is a 48 by 48 array. It's about 7 by 8 millimeters and in 3-micron technology. So this is a lot of stuff, about 100,000 transistors.

Now lets look at some of the circuits in more detail. Let me remind you again what it looks like. We're talking about a receptor that comes down into this synaptic arrangement, often called a triad synapse, which involves the horizontal cells, the bipolar cells and the receptor. The horizontal cells are connected together with little things called gap junctions, which are sort of high resistance connections. You can think of the whole thing roughly as a resistive sheet spread out laterally along the retina. The first thing I want to talk about is that model I showed you before. We make these resistors into this discrete, hexagonal array; and then we have a little triad synaptic arrangement which looks like this. This is a translation, a loose translation, of neural circuitry into MOS circuitry. I always tease the students that this is a bipolar cell made out of MOS transistors. The model we have of this triad synaptic arrangement is that it takes the difference of voltage between the horizontal cells and receptors. And that leads to a release of neuro-transmitter, which pulls the horizontal cells along toward the potential of the receptor. It also pulls the bipolar cell along in the process. We model that by this arrangement: These amplifiers are transconductance amplifiers, by the way. They have an output which is a current that's proportional to the difference in the input voltages, a very simple circuit in this technology. This is a little follower arrangement that

pulls on the horizontal resistive network in order to make it come nearer to the receptor potential. And we measure the current it takes to do that, which appears as a voltage between the receptor and the horizontal network.

It is a very simple model of the neural circuit. It's a very explicit model, though, and it makes some very explicit predictions which can be tested. The resistive network itself is also made out of such circuits. Of course, resistors don't come in MOS technology; you have to make them up out of transistors. It takes, on the average, three or four transistors to make a resistor, depending on how many are shared with each pixel. These share 6 with each pixel. We take 7 transistors to get started, and it takes 2 for each of the resistors. This is the point spread function, or the Green's function, or the spatial-impulse response, depending on whether you're an image processor, or a physicist, or an electrical engineer; they're all the same thing. To get it, you put a potential at one point in a resistor network and watch it die off as it goes out. This one happens to be a one dimensional network. It's a little more sharply peaked function for two dimensional networks like we have in the retina. This difference between the resistive spread in the network and the receptor potential is what's responsible for gain control. The potential of the receptors in the surround sets the potential of the resistive network, and any difference from that results in an output.

These are some real data. They show the response as a function of the light level of the center and as a function of the light level of the surround. You see you get the saturated curves just like the caricature we showed before. When you do that same experiment on our retina you get this kind of curve. Once again, this is the log of intensity vs the response; and indeed, the response moves over to keep itself centered on the intensity of the surround. Now, that simple level adaptation results in a very powerful signal processing function. If you take an edge and move it over a place in the retina, what you see is this second-derivative kind of response. It's a Laplacian filter, a smoothed version of a Laplacian filter. This is what our retina does when you do that same experiment. You get this nice second derivative.

Going further down to the next layer, you find this peculiar synaptic microcircuit, and we make an even looser translation from neural circuitry to silicon circuitry: we say that what this synaptic arrangement is doing is taking the difference between the signal and a temporally smooth version of the signal. And of course if you do that, the difference between the signal and the temporally smooth version looks like a time derivative. This is what it looks like in an animal; and this is what it looks like in one of our retinas. There are

various varieties that have sharper or smoother derivatives, depending upon the exact synaptic arrangement.

Once you have this system working this way, you notice that there are consequences, signal processing consequences, that come about just from the necessity of having the adaptation. That happens both in the horizontal layer and in the amacrine layer. Here you notice that this is now still up in the horizontal layer. If you put a flashing light on a whole retina, you get a very steep "on" response, and then the inhibition rushes in from the surround and shuts it down. On the other hand, if you have just a spot of light in the middle that flashes on, the inhibition doesn't rush in from the outside, and the output stays up. You can get that same kind of response with our retina. This is what happens with the light over a rather large area. You turn on the light, you get this steep increase, then you get the inhibition rushing in. If you have a small area, you get a much more sustained response. In fact, if you adjust the retinas slightly differently, you can make it completely sustained.

I'd like to show you a film of the retina in action. This is a picture of the set up. The retina itself is sitting in this 64-pin package. This is an old 16 mm movie lens sitting on top of it, focusing an image down on the chip itself. All this junk here is to remind the students in the audience that none of these things come packaged in pretty little boxes with nameplates on the sides. You always start out with some kind of "kluge" on a bench. But don't worry; they grow up. This chip is looking up at an image. I told the guy that works in the lab that I'd like to image one of the great big fans in the ceiling on the retina, and this is what he bought me. But its OK; it goes around like one. And so what the job of the retina is going to be is to look for time derivatives. We can slow the thing down, or speed it up, and we're going to see if we can model the kind of cells of the retina that look for time derivative signals, primarily for motion events. They're almost the entire population on the periphery of your retina, but there are not so many of them in the fovea, where you're looking for sharpness. You can see a little image here of the pinwheel when its not moving. This is a peculiar display. I used an ordinary oscilloscope with a "kluged" circuit that will displace the vertical position of the trace depending upon the signal. You'll see when it starts moving in a minute that the signal goes down. There it goes. So its looking at motion all right, and it doesn't even have a Cray computer behind it. You notice what looks like noise in the background there. That's not noise. The derivative machine actually has a little ringing in it, and that's the ringing of the derivative machine aliasing against the scan.

Now why would your eye bother to go and clip this big piece of machinery out so you have to swivel it around and do all that stuff? Well there's a good reason for it. And it is that, once you've time sampled the image, you've aliased away a lot of this motion information. Once you've got the motion information, you can turn it into nerve pulses in the optic nerve and its okay. But, it's not okay to sample first. What that would do is turn a real simple problem, namely taking a smooth time derivative, into a really hard problem. This is what people in artificial intelligence call a correspondence problem: trying to figure out what point in the second frame went with the corresponding point in the first frame. And that's a hard job. Well, I think we've seen enough of this thing running now.

There's a lesson in the retina. And the lesson I take home from the retina is the following. The nervous system works in real time, with real signals. And the one thing you don't ever want to do is to blank out. The nervous system has to stay operating under all conditions. Nature has solved this problem by working with differences, by developing a second order Laplacian filter. The time derivative is really there, if you want to think of it that way, to AC couple the system. Because it's a very high-gain system, you can't afford to be DC coupled all the way to the end, or you are going to be slapped against the top or against the bottom by the time you get to it. So what you do is transmit the differences. Once you've done that, you can see motion very nicely. And of course, then you sort out that. And the animals that did it that way ate the ones that didn't. And oh, by the way, if you start applying gain control to the derivative, you have the makings of a relative motion system. All because you've adapted to keep the system in range and off the stops. That's an important lesson, and it's one that we engineers tend to forget all too often.

Now what are really doing here? We're evolving a technology. And technologies evolve under a set of constraints. Those constraints are placed on us by the technology we work in, by the demands of the problems we try to solve, and by the sources of information that we are looking at. And I claim we need to learn from biology. Biology has a lot to teach us. Every time I understand another thing that it's doing, I say, " Ah, that's clever! I wish I had thought of that." Sometimes it takes me a long time to do that; I feel really stupid. And they are always obvious in retrospect. If we're going to learn from biology, we'd better be evolving our stuff under a set of constraints that have some similarity to the constraints under which biology evolved. I'm delighted to see the commercial exhibits downstairs. One thing everyone knows is that there's a lot of similarity between the evolutionary environment for business and for living species.

So what are the constraints for this technology? Well, I claim that wetware had constraints on wire. You wouldn't want to have to cart the brain around in a pickup truck, and power it with Boulder Dam; it just wouldn't do. And there's a constraint on specification complexity. I mentioned that we don't have enough genes to specify the whole brain. We sure don't have enough programmers in the world to program all our neural networks; that's for sure. What the nervous system came up with in a lot of places -- all the way from the retina up to the cortex -- is the two-plus-epsilon dimensional structure. The cortex is flat. Half of its thickness is white matter, wiring; half of it is where the processing actually goes on. That amount of wire looks like a lot compared to VLSI, until you realize that the cortex is a millimeter thick and a meter wide, or a meter square. So it really isn't anywhere near as many levels deep as it is in width. And that means its a basically two dimensional problem. And once you have a basically two dimensional problem, you're just worrying about numbers: what's the price of a wire. And there are things we can do about that. There is a highly local wiring strategy for unsampled data Things are computed in a context, and that context is invariably mapped into a physical locality in the nervous system. It's a nonlinear analog processor. Time is its own representation. And there are specialized long distance, sampled kinds of communications.

Well, I guess I've come here to bring you some bad news. The bad news is that we're going to have to learn how to design this stuff. I hope I've convinced you of that. I'd like you to start doing that. But it means some things. It means some things that are only remotely being thought about today. The locality issue is a central issue. Even if you're going to learn there's an underlying structure. In the brain, it's put there in the genes; in a chip it's put there with the wires that are built into the chip. And then you can go learn from there. But there's an underlying architecture. Normalization is the first order of business, both for level and for gain. And that's true in every level of the nervous system. And time-domain processing is everywhere. Time is an essential axis of every input, and of a lot of much higher-order processing. I think these are very general things about the nervous system. They don't have to do with just retinas, or anything like that.

I couldn't go away without giving you the good news. It's actually pretty good. It's good enough, I think, that I'm hoping I've convinced a lot of you that it is worth the pain. Silicon can achieve 10^8 synapses right now. Its going to be 10^{10} or 10^{11} within about the next ten years, the way the technology is evolving. And it's evolving without us having to push it. Its just clipping right along because of

microprocessors and memories and all that digital stuff. The devices have really nice computational primitives: exponentials, hyperbolic tangents,... all that kind of stuff. It comes very naturally from the technology. The analog representation isn't very precise, but you can get time constants over seven orders of magnitude if you like, so it has really good dynamic range. And it has nice continuous-time processing ability. Wafer scale integration is easy with analog technology, because built into it is micropower -- all these things running below threshold. They run much nicer down there. This whole retina runs in the microwatt range -- less than a milliwatt. That means that you can build a very robust system. There's an inherently current-limited behavior in all these circuits, so if one of them gets stuck someplace, it's not a big deal. And you can build time-multiplexed white matter by using the standard time multiplexed analog representation that you find in television sets. And you can build-in photoreceptors.

Now I realize that sometimes the good news is hard to swallow, because its a whole new technology, and it's kind of risky. But I'd like to remind you of something. Some of you here probably remember the wonderful poster that came out in engineering circles when the first transistors came along. It showed a great big vacuum tube running along and this poor little transistor scampering ahead of it upon three legs trying to get away. And the caption said, "HELP STAMP OUT TRANSISTORS!" It was wonderful. Well, maybe some people were serious about that, but a lot of it was tongue in cheek. And before long we were all out designing transistor circuits. And then the integrated circuit came along, and we said "Ah! They'll never replace <u>our</u> circuits! Ours are much better than that. We can wire them up on these circuit boards." And then they got cheaper, and they got better, and before long we were all wiring them into things. And then my friend Frederico, who's here at the meeting, came along with his first microprocessor, and we all said, "I don't want to learn how to program that miserable computer." And some of us didn't for a long time. And then they got better and cheaper, and now we're all programming all the time. I have a unique belief in the resilience of our engineering community when faced with new things. I think in one of the few places you can go in the world and find that new things are welcome, and that people want to learn about and use new stuff. And I believe that, 10 years from today, we're going to see a whole lot of neural networks built out of analog silicon.

Thank you.