# Computers

Some of you will remember Commander Data of the Star Ship Enterprise. He comes from the Second Generation series of the Star Trek films than ran on television for many years. He looks like us, but he is different in several important ways. First, he is man-made and in fact non-biological. If you were to open up his service panel, you would find mostly computer hardware. He is what Rene Descartes would call an automaton. Second, his mental powers are vastly greater than any modern computer you can imagine. Third, he has no emotions. Despite this, he is a competent member of the star ship. He always acts ethically; he claims that he was programed to do so. He is a consummate violinist, but he claims that he has simply memorized the playing of the great violinists of the past. He had a deep friendship with a woman despite the fact that he claims to have no idea what it means to fall in love. Data raises some fundamental questions for us. There is an ethical issue; should he be treated as a human being or as property? Then, is he conscious? So far as I know, no one ever asked him that question. If you were to ask him, I suspect that he would reply that he simply didn't understand the question. For our purposes, the big question is this: is Data possible even in principle? I suspect the answer is no, but the matter is far from settled.

Let me start the discussion by telling you two stories.

My first story concerns Charles Babbage. He was one of the intellectual giants of his day, a mathematician, philosopher, mechanical engineer, and an inventor. He was Lucasion Professor of Mathematics at Cambridge University, a position that was once held by Isaac Newton and is currently occupied by Stephen Hawking. He had one additional advantage over both these men: he was outrageously rich! Today we think of him as the father of the modern computer.

In 1822 he proposed to the Royal Astronomical Society a machine to calculate mathematical and astronomical tables. He called it the difference engine. It would have had 25,000 parts, weigh 15 tons, stand 8 feet tall, and be driven by steam. The British government gave him £1700 to start. By the time the government had lost faith in the project Babbage had received and spent £17,470 and the machine was nowhere near finished. About the same time, John Bull built the first steam locomotive for £784. Babbage had spent the equivalent of 22 locomotives! About the same time, Babbage had conceived of a better plan that would have been simpler to build with fewer parts. He called it Difference Engine #2. It was never built in his lifetime, but in 1991 the Science Museum of London built a Difference Engine from Babbage's original plans. It works flawlessly. It can evaluate 7th-order polynomials, store 8 numbers and calculate to 31 digit accuracy.

In 1837, Babbage proposed a mechanical general-purpose computer as the successor to his difference engine. It was called the Analytical Engine. It incorporated an arithmetic logic unit, control flow in the form of conditional branching and loops, and integrated memory, making it the first design for a general-purpose computer. In modern terminology, it was Touring-complete. This means roughly that in principle it could calculate anything that any other computer could calculate. It was not until the 1940's that the first general-purpose computers were actually built, more than a century after Babbage had proposed the pioneering Analytical Engine.

Let's switch to modern times. In 1997 IBM built a computer called Deep Blue exclusively to play chess. It could evaluate 200 million chess positions per second and used as much electrical power as a small village. It played a match with Gary Casperov the reigning world champion and lost by a score of 4-2. The computer was then massively upgraded and won the next match by a score of 3 ½ - 2 ½. Casperov accused IBM of cheating, i.e. using human players. IBM denied the charge. Casperov demanded to see the computer program. IBM refused and promptly dismantled the computer. So did IBM cheat? Probably not, but that is not the point. The point is that a human player who could evaluate perhaps two positions a second could beat a computer that could evaluate 200 million. How is that possible? The easy answer is that they were the right two, but how did he know?

Let's start by thinking of the human brain as a kind of computer. In a previous talk I told you a little about neurons and how they work. I described them in terms of electrical engineering and chemical engineering. Now since we are talking about computers, let's look at neurons from the point of view of mathematics. The header to this talk is the sketch of the mathematics of a neuron. According to the electrical engineer, electrical signals are propagated through the dendrites to the body of the neuron. From the point of view of mathematics, they are just "quantities." In the diagram they are labeled $x_1$, $x_2$,⋯,$x_m$. Some of these quantities are larger than others, so we will indicate this with "weights," here labeled $w_1$, $w_2$, ⋯,$w_m$. The reason some are larger than others is a result of the chemical calculations carried on by the synapses; in the diagram they are labeled synaptic weights. It is speculated that memories are stored in this way. When you remember something you are just summoning up a constellation of synaptic weights! Then these quantities are just added; that's the significance of the big sigma in the diagram. Depending how large the sum is the neuron either does or does not send a signal onto the next neuron. This final decision is made by the activation function, here represented with a phi. That's all there is to it – just high school algebra, and high school algebra is something that computers are very good at. Perhaps we can make simple "brains" by running networks of these mathematical neurons on computers.

There are philosophers who would like to take this further. They claim that the brain is nothing but a very complicated computer. This point of view is called connectionism. If this were true, if this were the whole story, there would not be much for the philosophers to do. We are conscious in the same way that computers are conscious. We have free will to the same extent that computers have it. We understand computers, after all, we build them, so by extension we understand our minds. You have all heard of the branch of computer science called artificial intelligence or AI for short. Is there really any important difference between artificial intelligence and our own? To explore this question, I would first like to give you some history of computers and artificial intelligence, then tell you about special kinds of computes called neural nets, and finally get back to the big philosophical questions.

Calculating machines, for example Babbage's analytic engine, are not a new idea, and when electronics with switches and vacuum tubes were first developed it became possible to make circuits that could calculate things. They had something in common with the mechanical calculators, however. If you wanted your calculating machine to calculate something different you would have to rebuild it. If you wanted your calculating circuit to calculate something different you would have to rewire it. The modern computer was born in 1945 with the idea that a calculating circuit could be made in some sense to rewire itself. The man who first envisioned this, the man who is usually given credit for inventing the modern computer was John von Neumann.

John von Neumann was born in Budapest in 1903 and moved to Princeton with his family in the mid 1930's. By the mid 1940's he had invented game theory and the theory of automata (which deals with the possibility that machines might be able to reproduce themselves). In the field of hydrodynamics, he developed a mathematical formalism for modeling shock waves.  These calculations were used in the Manhattan Project to develop detonators for nuclear weapons. He was also the author of the pivotal book on the mathematical foundations of quantum mechanics.

A von Neumann computer consists of two parts, a memory unit and a central processor or CPU. The memory stores two kinds of information: data in the form of binary numbers and instructions that tell the CPU what to do with these numbers. The computer operates in well-defined cycles: first fetch some specific data into the CPU together with instructions that tell the processor what to do with these numbers. The results of the calculation are then stored back in memory and the computer begins a new cycle. This is called the von Neumann architecture, and such a computer is said to be a serial machine; serial in the sense that it only does one thing at a time. This has been a successful strategy because modern computers are so phenomenally fast. There is something about it that seems wildly impractical, however. Suppose you were moving all your household possessions from New York to San Francisco. The von Neumann architecture corresponds to moving them one item at a time; move a dress to San Francisco, then go back to New York to get a coat, etc. The speed at which signals travel in a computer is limited by the speed of light, 186,000 miles a second. The speed at which signals travel in the brain is more like 100 yard, the length of a football field, in one second. Clearly the brain is not a serial machine, but remember there are one million ganglion cells in the optic nerve. In some sense the brain is doing one million things at once. This difference has motivated the development of neural net computers, which I will discuss eventually.

Break here - - - - - -

The second giant in the field of computing and artificial intelligence was the Englishman Alan M. Touring. In the field of computer science, he has somewhat the same status that Einstein has in physics. He is the towering giant of the 20th century.

During World War II Touring was instrumental in cracking the extraordinarily complex code that the Nazi military used for most of its communications. The simplest code you can imagine is simply a set of rules for replacing one letter for another. For example, every time the letter T appears replace it with Q, replace U with B, etc. If your correspondent has a copy of the rules, he or she can easily translate the coded message. Unfortunately, your enemy can easily reconstruct the rules based on the frequency with which letters appear in English text. But suppose the rules change with each successive letter. It seems that that would make the code impossible to break. The Nazis managed to write such a code with the help of a mechanical apparatus called the *enigma* machine. This was an electro-mechanical rotor device that would change the code automatically. The sender would encode the message with an enigma machine and the recipient with another machine would decode it.  With some of the more complicated enigmas there were billions of billions of possible combinations.  Even if you could review a million combinations a second, it would still take millions of years to decode any message, and the Nazis changed the machine settings every day! Part of the British success in breaking the code exploited regularities in the German communications. For example, they tended to end their messages with "Heil Hitler." There was also a weakness in the machine in that it could never code a letter as itself.

A major part of cracking the Enigma code made use of another machine built by Touring called the *Bombe,* which could sort possibilities with immense speed. Thanks to Touring's work, the Allies were able to decode almost all intercepted

Nazi messages throughout the war. This had its downside; it forced some agonizing decisions. For example, the British knew that the city of Coventry was to be bombed, but they decided not to evacuate for fear of revealing the fact that the German messages were being decoded. Nonetheless, it could be argued that without Touring's genius, the Allies would have lost the war or at least it would have continued much longer with great additional suffering on the part of the British people.

The story does not have a happy ending. Touring was an atheist and a homosexual and he was not discreet. Homosexual sodomy was treated as a crime in Britain in the 1950's. In the course of reporting a burglary in 1952, he made the mistake of informing the police of his homosexuality. He was arrested, tried, and convicted. As part of his sentence he was to be subjected to chemical castration. It seemed inevitable that his security clearance would be revoked and he would be subjected to public humiliation. In June 1954 he committed suicide by eating an apple dipped in cyanide. He was just 42.

For our purposes, Touring is notable for two other contributions, the Touring machine and the Touring test. The Touring machine is a sort of a mathematician's abstract computer. No one would every build a computer like it, but because of its extreme simplicity it is possible to prove rigorous mathematical theorems about what computers can and cannot do. The machine consists of an infinitely long tape marked off in "squares" on which are written symbols like 1 and 0, and a read head that can read, erase, and write symbols and move the tape one square to the right or left. The read head contains a set of instructions of the form:

> If the head is in state 17 and the square contains a 1, change to state 43 and move the tape one square to the right.
>
> If the head is in state 78 and the square contains a 0, replace it with a 1 and move the tape one square to the left.

Nowadays we would say that the tape was the "data" and the instructions in the read head were the "program." Touring was able to prove that any calculation that could be performed on any computer could be done by this simple machine. He also noted that there are calculations that seem quite trivial to humans but which cannot be done by any computer.  This last point is worth some careful thought. Computer operations are said to be “algorithmic,” that is to say that the computer’s calculations are always done according to a list of instructions. When we are performing operations in our head that are impossible for any computer we are thinking non-algorithmically. It has been claimed (and also denied) that our minds are ultimately not computers and that computers will never be able to think like us because we are capable of non-algorithmic thought. Touring himself didn’t think so; he claimed that mathematical arguments are no help in deciding if a machine can think. He proposed instead an experimental test, which he called the imitation game and which has come to be known as the Touring test.

Imagine that you are communicating with either a computer or another human being through a computer terminal. You submit questions and the unknown entity replies. Your job is to decide whether you are communicating with a person or a computer, and of course, the computer doesn’t have to tell the truth. So for example, if you ask the computer, “Are you a computer?” it will reply “No.” An intelligent computer, claimed Touring, is one that can systematically fool the human questioner. The implication is that this is the only meaningful way to define intelligence.

So how well are we doing in building such a computer? There is a contest held every year called the Loebner Prize. Contestants submit their computer programs, which are questioned by a panel of twelve judges. If you can fool the majority of judges for five minutes you win $100,000 and a gold medal. No one ever has. There’s actually a website where you can go and interview some of the best programs from past competitions.

http://www.loebner.net/Prizef/loebner-prize.html

The Touring machine is an example of algorithmic computing, i.e. it operates according to a definite set of instructions. The instructions are contained in the read heads and are put there by the people who build the computer. Another way to look at it is that the Touring machine works by manipulating symbols, in this case the symbols 1 and 0. All von Neumann style computers are symbol manipulators. The symbols now are represented by 32- or 64-bit words, but they are symbols nevertheless. In the early days of AI it was assumed that intelligence could be reduced to symbol manipulation, and this approach has come to be called Good Old Fashion Artificial Intelligence or GOFAI for short. This approach had some notable successes early on. Programs were developed that could prove theorems in logic, mathematics, and geometry, and play decent games of Backgammon, checkers and chess. In retrospect, these tasks are most easily represented by symbols. Other tasks, face recognition for example, proved to be much more difficult. Other aspects of intelligence, like pattern recognition, context sensitivity, and rough-and-ready categorization proved much more difficult. It became increasingly clear how few human abilities actually fit the GOFIA model. Certainly the brain in no way resembles a von Neumann computer. The brain does parallel processing on a massive scale and it does it with neurons. So why not model a computer like the brain? Thus was born the idea of neural nets. Connectionism and parallel processing are more or less synonymous terms.

The header of this podcast is a very schematic diagram of  simple neural net. The circles represent neurons and lines connecting them are a combination of an axon from the previous neuron, a synapse, and a dendrite leading into the next neuron. Associated with each synapse is a weight. This is a small number which might be positive or negative. (In this diagram we can simply say that the weight is associated with the entire line.) The weights can be adjusted, and in this way the network “learns” from experience. From the point of view of mathematics, a neuron is just a function called the activation function. It simply adds up the incoming signals, and depending on how large the sum is it produces some output that is fed onto the next neuron.

Suppose we want to teach a network to recognize faces. First we digitize some photos so that they can be fed into the net. We know the names of these people of course, and the network has a list of possible names. We chose a sample of photos called the training set, and feed them into the net. The net has never seen these people before, and so its responses are just wild guesses. We now change the weights so that it makes better guesses. Feed the training set in again, and again adjust the weights. We might have to do this many times until the network is completely trained. How well does this work?

Here's an example in some detail. Garrison Cottrell and his group at UC San Diego constructed a three-layer neural net that used photos as input. The training set consisted of 64 photos of 11 different faces with different expressions plus 13 photos of non-faces, photos of a watch or a bulletin board, for example. Once trained, the machine achieved 100 percent accuracy on the training sample. It could tell you which was a face, what the sex of the person was, and who it was. The team then showed the computer many picture of the same people but with different expressions. It was 98% accurate at identifying the name and gender of the new photos. The team then showed the net more pictures of people it had never seen. It was still 100% accurate in distinguishing between faces and non-faces, and 81% accurate in determining the sex of these people it had never seen. I think that if you showed me photos of faces of students on campus I would not do much better myself.

There are some things about this I find fascinating. If you were to stop your desktop computer in the midst of a calculation and peer into its CPU you could discern exactly what it was doing. Look into its program memory and you will learn what it is going to do. Look into its hard drive and you will see the data it is working with; but with a neural net you never know what it is thinking about. The result of all its knowledge and judgement is a set of apparently meaningless numbers.

This brings me around to the question I started with; is the mind nothing more than a large computer? We know now that it is certainly not a von Neumann style computer, but might it be just a very large neural net? It is certainly true that we can model or instantiate various mental processes or sub-processes with artificial neural nets. It is also true that when we think about something, various areas of our brains “light up” in fMRI scans. To use a technical term, there are “neural correlates” to our thoughts. But is that the whole story? People working in computer science will often say glibly, yes of course, that is all there is. Philosophers are not so sure. When philosophers argue the issue, it is usually with the help of some bizarre thought experiments.

The most famous of these is called the Chinese room argument first proposed by John Searle of UC Berkeley. Let's assume that you don't know any Chinese, but you are in a room with many instruction books written in English and Chinese. Chinese speakers pass questions into you written in their native language. You then consult your reference books as to how you are going to answer. They say things like, "if you see the following set of meaningless symbols, reply with this set of equally meaningless symbols." You pass your responses on to the Chinese speakers outside, and they assume you are also a native speaker, but in fact, you understood absolutely nothing you wrote.  Searle's point is that in the room you are doing exactly what a computer does: you are taking input data and processing it according to a set of instructions, but without any understanding of what you are doing. The input, the output, and the instructions are all meaningless. Searle makes the following distinction. Syntax refers to the rules for making grammatical sentences. Semantics refers to the meaning of these sentences. Computers are masters of syntax; they know all the rules, but they know nothing of semantics. It would be trivially wrong to say that our minds are not computers. We can compute after all; but our minds are not only computers. We have semantics, they don't.

Searle makes a further point that seems to me very important. Suppose you are typing a letter using MS Word. You are communicating with the computer by means of electrical impulses. The computer responds by putting little black patterns on your monitor screen. Searle's point is that neither the electrical impulses nor the letters on the screen by themselves have any meaning. The meaning is intrinsic to you, not the computer. Paper money makes a good analogy. Think of a hundred-dollar bill. It's just a piece of paper. In and of itself it is practically worthless. We assign it meaning and value.

There have been numerous attempts to refute this argument. Perhaps the best we can do by way of a refutation is to say that Searle's argument is based on our current knowledge of computers and consciousness. Perhaps one day we will see all this from a larger perspective. Perhaps. I would not want to discourage further research in computers as means for understanding the mind, but there are further philosophic considerations that I will take up in the next podcast.