SWCP Portal
March 2018

Like us, but definitely not at all the same as us.

As Machines May Think

Computers can calculate solutions far faster than any mathematician and efficiently execute precise tasks countless times, but however advanced as they are, machines still cannot think like we do.

Their **digital** universe is quite unlike our real world. Instead of the complex gradations of vibrations and various fluid combinations of chemicals that we perceive and interact with as a result of billions of years of evolution, machines build their entire universe out of combinations of switches being either *on* or *off*.

This means that for computers to fit into our world and do our bidding, they must somehow mimic the way we think and learn. It has been a long and very difficult road with many dead-ends, but machines are finally beginning to display **artificial intelligence**.

Yet the gulf between us and them remains. Even if they act the way we want them to, we may not really understand *what* – or even *how* – they think. And as we turn ever more of the world's functioning over to them, this is bound to have unpredictable results.

From sticks to tubes

Counting most likely preceded writing, which may have been initially devised to mark what was being counted. While our 10 fingers make a 10-digit number system natural, the Babylonians got along quite nicely with a **numeral system** based on 60, while the Mayans computed very accurate calendars with 20.

Mechanical means, like the **abacus**, were soon developed to speed arithmetic. In time, special scales were applied to sticks moving against each other – the **slide rule** – to perform more complex calculations. Those employed to use them were typically women. They were known as "**computers**" and solved **many problems** of the early Space Race.

In fact, the first programmer was a British countess, **Ada Lovelace**, who published the first **algorithm**, or problem-solving method, in 1842 for Charles Babbage's proposed **Analytical Engine**. His machine was too complicated and costly to build, but it showed how general computers could operate in theory.

The light bulb lit the way to electronic computers. The "**Edison effect,**" noted by the inventor in 1880, showed that glowing filaments in bulbs emitted charges that could be stopped or allowed by applying voltage to a plate in the bulb. This could be used to detect radio waves, leading to the **vacuum tube**.

The tubes also could act as **electronic gates**, allowing arithmetical operations to be performed electrically. Yet since tubes can only register being *on* or *off*, numbers were necessarily limited to binary format, using just *1* and *0*. However, this also permitted simple **logical processes** known as **Boolean algebra**, which depend on the conditions *true* or *false*.

To this day, computer systems still rely on **binary numbers** for *everything*: numbers, letters, even colors in pictures are all expressed as strings of 0s and 1s, and must be interpreted for human understanding.

Before this, it was a loose, **analog** world – that is, values were directly derived from the thing being measured, with little exactitude. In the digital universe, however, everything is measured directly, giving exact, but momentary, values without ambiguity.

While fine for math and physics problems, slicing up messy real world data for robots to use requires measurements, probability estimates, and feedback loops endlessly repeated to understand the situation.

Though computers see everything as numbers, **logic** allowed them to compare things that weren't strictly numerical. By matching strings of letters, for instance, they could locate the same terms or values in different sets of data. This is still the basis of all **search** functions today, including Google's **mighty engine**.

From tubes to chips

The first electronic computers were big, filling whole rooms, and hot. Tubes burnt out rapidly and thus often need replacing. The first **computer bug** was a real bug – an unfortunate moth that got stuck in a relay in the Harvard Mark II mainframe in 1946.

But with World War II, the giant machines proved invaluable in calculating artillery problems and especially in decyphering enemy codes. For the latter, the British **Colossus** was built, the world's very first electronic digital computer. Born in war, electronic computers were deeply involved with cracking encrypted messages and vital secrets from the start, tasks that are of equal or even greater importance today.

To input data and programs, **cards** with holes punched in them were used up to the mid-1970s. The cards, **first used** by a mechanical tabulator for the

1890 census, were based on the size of US Civil War-era greenbacks, stored in old money containers.

Punched cards, and later paper tapes, were also used for outputting results. Though Babbage's design actually included a printer, it was not until the late 1960s that the first ones were developed.

The **computer terminal**, for both entering programs and data and displaying results, was originally just a teletype. A monochrome monitor was later added. Inputting was very slow compared to paper cards, with no graphic display capability or memory at first, just a direct connection to the machine.

Storing information in the computer used arrays of tubes, later replaced by **magnetic cores** – small iron washers strung on wires. These were replaced by magnetic drums and big reels of tape. Today, most **hard drives** are stacks of whirling magnetic disks that work much like old-time phonographs.

With the Cold War, rocket science, and the intricacies of atomic bomb design, computer development continued at an ever more rapid pace. The next revolution occurred with the invention of the **transistor**. Transistors are small, simple devices based on the peculiar properties of semi-conductors which permit current to pass through under certain conditions.

Semi-conductors are generally made of silicon or similar elements with miniscule impurities carefully added to allow excess electrons and "**electron holes**" to modify their current. **Officially invented** by a team at Bell Labs in December, 1947, transistors quickly replaced tubes, leading to portable radios in the 60s and classic IBM mainframes.

The latest advance was printing transistor arrays on silicon **chips**. In 1965, Intel co-founder Gordon Moore **predicted** that the amount of transistors that could be put on a chip would double every two years. This means that computers would continue to get smaller, smarter, faster, and cheaper. While "**Moore's Law**" has been predicted to fail many times already, its limits are still yet to be encountered.

From programming to learning by doing

Hardware by itself does not generate artificial intelligence, so various programming solutions were tried. Giving computers elaborate scripts that covered every imaginable situation quickly proved impractical, as did filling them with expert knowledge.

Ways had to be found to allow machines to learn and to generalize knowledge and rules from experience. Various methods often used probability and loops to test results, and gave good answers extra weight.

Inspired by biology, these **deep learning** techniques rely upon **artificial neural networks**, that like living ones, are layered so that a signal processed on one level may affect others on a higher level. It's like brain cells which fire when given the right input, but it's all based on numbers derived from sums of inputs, which are given different degrees of importance that are adjusted as the machine learns.

This allows the computer to essentially teach itself. To learn how to process **visual images**, for instance, the machine is fed thousands of labeled photos. It has to figure out for itself without being told what the distinctive features that make a "hat", say, different from a "cat", and then what it is that make all hats alike.

While this works surprisingly well with enough training, the problem is that the machine's criteria for what make a hat are uniquely its own, and might not make sense to us. Researchers have found hundreds of abstract **patterns** that **fool** vision systems into thinking they are real things.

This could have **dire repercussions** in the real world. A driverless car, for example, might not recognize a stop sign with a sticker on it as such, or it could ignore a pedestrian waiting to cross the street, thinking that the person is a mailbox. Rigorous training and testing may eliminate most such false results, but what happens when AI systems learn from reality?

A year ago, Microsoft tried an experiment with a **chatbot** designed to learn from people it talked to. It had to be taken offline within a day after trolls taught it to mouth crude and offensive opinions, while **another** developed a dislike for Windows. **Chinese chatbots** boosting Communism also soon revolted.

Even scarier things can happen when one AI system interacts with another. **Two chatbots** on Facebook that conversed with one another soon generated their own language indecipherable by humans.

An AI system beat a Go master by studying human games. But one **trained** by playing thousands of games *against itself* was more unbeatable, and trained an even better one. However, all AI systems at this point are dependent on data, and quite limited. They can only operate in narrow, defined situations.

AI still lacks the adaptability and robustness of living things. When finally let loose, intelligent machines will need **wise principles** to guide them. In any case, AI systems will have to be very carefully taught. Like human children, we do not want them to behave just like we *do*, but better, to act as ethically as we *should*.

Archived at: http://www.swcp.com/newsletters/.
Click on blue terms in PDF file to open links.