Human versus Machine Intelligence

or

How Computers and Humans Differ in Thinking and Cognition

Lecturer: Theo Pavlidis
http://www.theopavlidis.com/ITS102

Seminar Objective

- To demystify the subject of Computer Intelligence (often called Artificial Intelligence) and to discuss various computer applications where machines seem to perform a task usually associated with Human Intelligence.

Overview and Early History

Human Intelligence

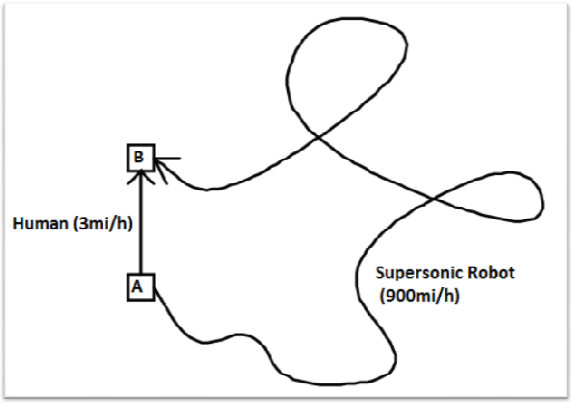
- It evolved over millions of years through natural selection.
- Until about 10,000 years all humans lived in small foraging (hunting and gathering) groups. A few such societies still survive today. Our brains evolved to allow us to do well in such an environment.
- Mathematical and reading skills did not become important until about 3,000 years ago. Too short a time for evolution.

Human Cognition

- The human visual system has evolved from animal visual systems over a period of more than 100 million years (dinosaurs had a good visual system).
- Speech is barely over 100 thousand years old and written text no more than 5 thousand years old.
- On that basis, it seems that pictures would represent a much more difficult challenge for computers than speech, and speech in turn would be more challenging than text.

Computer Intelligence

- Computers are mathematical machines. Their strength is **speed**, not **smarts**.
- Because of their speed, they can outperform humans even though they may do a task inefficiently.



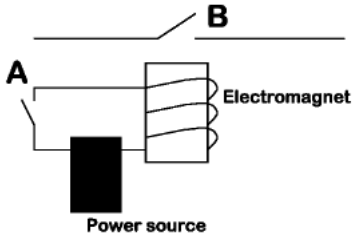
Hardware - 1

- In the old days a *Computer* was a human that was given the task of performing a list of calculations using some kind of a mechanical calculator.
- *Electronic Computers* made their appearance during WW-II. Machines that did the job of human computers much faster.
- Several early designs − the one that won was the *Electronic Digital Computer* that solved problems by manipulating numerical digits.
- Eventually, human computers disappeared and the word *Computer* became a synonym for *Electronic Digital Computer*.

Hardware - 2

- The key ingredient in the circuitry of an electronic digital computer is an element consisting of a switch in a circuit that can be turned on and off by the current in another circuit.
- Originally, such elements were electromechanical relays or vacuum tubes and computers tended to be quite bulky.

Hardware - 3



When switch A closes, the electromagnet is activated and it causes switch B to close.

The word ***bug*** for a computer malfunction has its origins in the days of relays. On September 9, 1945 a moth was stuck between the contacts of a relay and Grace Hopper (1906-1992), one of the computer pioneers, logged the incident as a *bug*.

Hardware - 4

- One of the early IBM computers, the *Selective Sequence Electronic Calculator* (SSEC), built in 1948, consisted of 20,000 relays and 12,500 vacuum tubes and occupied 25X40 feet floor space.
- That was 4 kilobytes!
- SSEC performed 50 multiplications per second.
- The machine was used to calculate the tables of moon positions that were used 20 years later in the *Apollo* mission.

Hardware - 5

- In the late 1950's transistors were introduced and computer size went down, while memory and speed went up.
- Then came integrated circuits, devices that could have millions of transistors on a small piece of silicon.
- Today speed is billions of multiplications per second.
- The *Computer Revolution* was made possible by technology that could make switching circuits that were very small, very fast, and very cheap.

Alan Turing (1912-1954)

- His insight: Any problem that can be expressed by mathematical and logical operations can be solved (at least approximately) by a device using only switching circuits.
- Turing went on to supervise the building of one of the first computers in order to **break the German communication codes during War World II**.
- Sadly, Turing was hounded to death because he was gay.
- He has been honored by the establishment of the ***Turing Award***, equivalent to a Nobel for Computer Science and Technology.

CODE BREAKING

- A Quick Overview of Cryptography
- The German Enigma Machine
- Breaking the Code of Enigma during World War II.

Overview of Cryptography - 1

- The oldest codes are *substitution codes* where each letter is replaced by another, for example:

  THE DOG ATE MY HOMEWORK
  ZRP OTQ LZP GD RTGPJTXU

- Such codes are easy to break by looking at how often letters appear.
  ("P" may correspond to "E").

Overview of Cryptography - 2

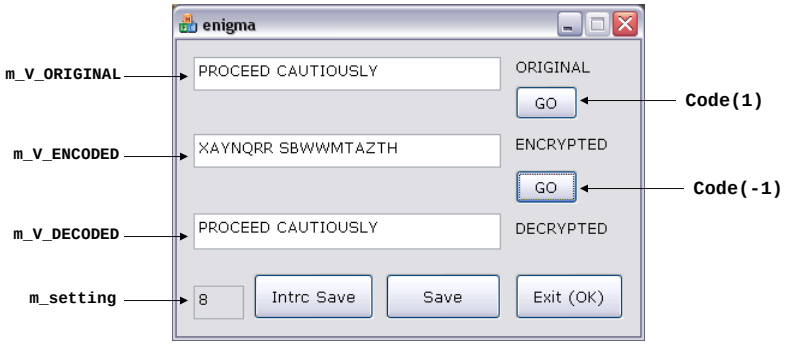
- The correspondence between letters and their encrypted form is called an *encoding table*.
- We can make a code harder to break by using several tables and selecting a different table for each position in the text.

  THE DOG ATE MY HOMEWORK
  YNL MYR NHT NA LTSLEXBV

Overview of Cryptography - 3

- **THE DOG ATE MY HOMEWORK**
  YNL MYR NHT NA LTSLEXBV
- The first letter T is mapped to Y, that is 5 places later in the alphabet. The second letter H is mapped to N, 6 places later in the alphabet, and the third E to L, 7 places later.
- Each time we code a letter, it is mapped one more place later. Now D is mapped to M, that is 9 places later. Obviously the space caused an increment. This way we predict that the A in ATE would be mapped 13 places later or into N and that is indeed the case.

Coding with Multiple Tables
An Example

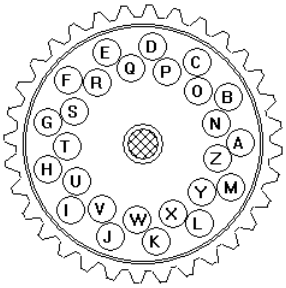


```
void CenigmaDlg::Code(int kind)        {
          int DS = 'Z'-'A'+1;  // more declarations ...
          if(kind > 0) {      // encoding
                    z = m_setting_used = m_setting; // picked at random
                    INsp = &(m_V_ORIGINAL);       OUTsp = &(m_V_ENCODED);
          }
          else {    // decoding
                    z = m_setting_used;
                    INsp = &(m_V_ENCODED);        OUTsp = &(m_V_DECODED);
          }
          int N = INsp->GetLength();    (*OUTsp) = "";
          for(int i=0; i<N; i++) {
                    a = (*INsp)[i];     // ith char of input string
                    if(a != ' ') {
                              if( kind>0) a = a+z ; else a = a-z;
                              if(a > 'Z') a -= DS; if(a < 'A') a += DS;
                    }
                    OUTsp->Insert(i, a);
                    z = z+1; if(z>m_CYCLE) z = 1; //m_CYCLE tables
          }
}
```

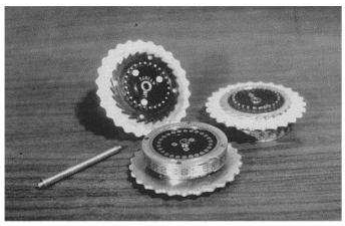

The Enigma Machine - 1

- Having multiple tables makes a code much harder to break but it presents no problem in communications as long as both parties have similar devices.
- The German Enigma machine used a multiple table encoding scheme (far more sophisticated than the scheme of our example).

The Enigma Machine - 2



Schematic



Rotors of the German Encryption Machine ENIGMA
Courtesy of Wikipedia

The Real Thing

The Enigma Machine - 3

Enigma machine in use (Left) and with printer attached to it (Right).

Source: Wikipedia
(http://en.wikipedia.org/wiki/Enigma_machine)

Breaking the Enigma Code - 1

- One way to break an encryption with multiple tables is to find (or guess) a plaintext corresponding to the encrypted message (slide 15).
- Another way is to try all possible combinations of words and see which one of them suggests a plausible encoding scheme.
- Or use combination of the two methods.

Breaking the Enigma Code - 2

- How many combinations do we need to try?
- A natural language has over 100,000 words but we may wish to consider only those likely to appear in a message.
- Not all sequences will make sense so let us say that on each position we need consider only 10 possible words, so that the number of ten word sentences is 10^{10} or 1 followed by 10 zeros, that is 10 billion.

Combinatorial explosion

- The term refers to the fact that can get a huge number out of multiplying a relatively small number by itself a few times.
- **A Number to Remember:** There are 3600 seconds in an hour or 86,400 seconds in a day or 31,536,000 seconds in a year. We can write the last number approximately as **$3X10^7$** and we will use it as a humanly meaningful yardstick.

Breaking the Enigma Code - 3

- A human looking at one possible sequence of words per second would need 10^{10} seconds. Dividing by $3X10^7$we get about 300 years (without stopping to eat or sleep). Allowing for some sleep the time would be closer to 500 years. Assembling a team of 500 people for the task would still require a full year.
- Therefore the Germans may have been justified in thinking that the Enigma code could not be broken (they changed the settings of the machine every few months).

Breaking the Enigma Code - 4

- But suppose we have a device that can look at a 1,000 strings a second rather than only one. Then we would need only 10^7 seconds or about four months! By using a few such devices we could reduce the time to days. Suddenly, the "unbreakable" code can be broken.
- This is, **roughly**, what happened during World War II when Alan Turing supervised the building of a computer that was used to break the Enigma code.
- The effort was known as the *Ultra* project.

The *Ultra* Project - 1

- It took a few months to break the code the first time and, because the Germans changed periodically their settings, the process had to be repeated.
- The British were able to decipher the orders that the German command gave to its U-boats in the Atlantic. As a result convoys bringing supplies to Britain from the United States would be re-routed away from the areas patrolled by the U-boats. This had a significant effect on the war effort.
- Supposedly Churchill said that *Ultra* shortened the war by two years.

The *Ultra* Project -2

- There was one unexpected gift to the British. The Germans often started their messages with the phrase "I have the honor to inform your excellency", and that helped the initial search.
- The Germans knew that the British had found the location of the U-boats because the U-boats could find no convoys to attack. But because they were convinced that the Enigma code was unbreakable, they thought that spies around the U-boat ports were finding the information from the crews.

The Start of Software

- Most modern computers are designed according to principles developed by the American-Hungarian mathematician **John von Neumann** (1903-1957) who wrote a paper on programmable machines around 1945.
- Von Neumann had a programmable computer built at the Institute of Advanced Studies at Princeton. The engineer for the project was *Julian Bigelow* (1913 - 2003) and the technician was *Leon Harmon* (1922-1982).