Chap 1 : History of AI

1.1

Intelligent machines, or what machines can do.

Philosophers have been trying for over two thousand years to understand and resolve two big questions of the universe:

How does a human mind work, and can no-humans have minds? However these questions are still unanswered.

The nature of philosophy allows for disagreements to remain unresolved.

However, engineers and scientists have already built machines that we can call intelligent.

Dictionary definitions

1. **Someone's intelligence is their ability to understand and learn things.**
2. **Intelligence is the ability to think and understand instead of doing things by instinct or automatically.**

According to the first definition, intelligence is the quality possessed by humans. But the second definition gives some flexibility.

The goal of AI as a science is to make machines do things that would require intelligence if done by humans.

One of the earliest and most significant papers on machine intelligence, "Computing machinery and intelligence", was written by Alan Turing in 1950.

Turing did not provide definitions of machine and thinking, he just avoided semantic arguments be inventing a game, the *Turing imitation game*.

Instead of asking, "Can machines think", Turing said we should ask, "Can machines pass a behaviour test for intelligence".

He predicted that by the year 2000, a computer could be programmed to have a conversation with a human interrogator for five minutes and would have a 30% chance of deceiving the interrogator that it was a human. A computer passes the test if interrogators cannot distinguish the machine from a human on the basis of the answers to their questions.

The imitation game proposed by Turing originally included two phases. In the first phase, the interrogator, a man and a woman are each placed in separate rooms and can communicate only via neutral medium such as a remote terminal. The interrogators objective is to work out who is the man and who is the woman by questioning them. The rules of the game are that the man should deceive the interrogator that he is the woman, while the woman has to convince the interrogator that she is the woman.

In the second phase of the game the man is replaced by a computer programmed to deceive the interrogator as the man did. It would even be programmed to make mistakes and provide fuzzy answers in the way a human would.

If a computer can fool the interrogator as often as the man did, we may say this computer has passed the intelligence behaviour test.

The interrogator may, for example, ask both the human and the machine to perform complex mathematical calculations, expecting that the computer will provide a correct solution and will do it faster than the human. Thus the computer will need to know when to make a mistake and when to delay its answer. The interrogator also may attempt to discover the emotional nature of the human, and thus, he might ask both subjects to examine a short novel or poem or even painting. Obviously, the computer will be required here to simulate a human's emotional understanding of the work.

The Turing test has two remarkable qualities that make it really universal.

- By maintaining communication between the human and the machine via terminals, the test gives us an objective standard view on intelligence. It avoids debates over the human nature of intelligence and eliminates any bias in favour of humans.
- The test itself is independent from the details of the experiment. It can be conducted either as a two-phase game as just described, or even as a single-phase game in which the interrogator needs to choose between the human and the machine from the beginning of the test. The interrogator is free to ask any question in any field and can concentrate solely on the content of the answers provided.

Although modern computers cannot pass the Turing test, it provides a basis for verification and validation of knowledge-based systems. A program thought intelligent in some narrow area of expertise is evaluated by comparing its performance with the performance of a human expert.

From a practical point of view, an intelligent machine should help humans to make decisions, to search for information, to control complex objects, and finally to understand the meaning of words.

To build an intelligent computer system, we have to capture, organize and use human expert knowledge in some narrow area of expertise.

1.2 The history of AI from "dark ages" to knowledge based systems.

1.2.1 The "dark ages" (1943-1956)

McCulloch and Walter Piters proposed a model of artificial neural networks where each neuron was postulated as being in binary state.
The neural network model stimulated both theoretical and experimental work to model the brain in the laboratory. However, experiments clearly demonstrated that the binary model of neurons was not correct.

In 1956 John McCarty, Martin Minsky nad Claude Shannon organized a summer workshop at Darthmouth College. They brought together researchers interested in the

study of machine intelligence, artificial neural sets and automata theory. This workshop gave birth to a new science called artificial intelligence.

1.2.2 The rise of AI, or era of great expectations (1956 late 1960)

One of the most ambitious projects was the General Problem Solver (GPS) (Newell and Simon, 1961, 1972)

GPS was the first attempt to separate the problem-solving technique from the data. It was based on the technique now referred to as means-ends analysis.

Newell and Simon postulated that a problem to be solved could be defined in terms of states. The means-ends analysis was used to determine the difference between the current state and the desirable state or the goal state of the problem, and to choose and apply operators to reach the goal state. If the goal state could not be immediately achieved from the current state, a new state closer to the goal would be established and the procedure repeated until the goal state was reached. The set of operators determined the solution plan.

However, GPS failed to solve complicated problems.

The computers were very limited.

1.2.3 Unfulfilled promises, or the impact of reality (late 1960 – early 1970)

From the mid-1950s, AI researchers were making promises to build all-purpose intelligent machines on a human scale knowledge base by the 1980s, and to exceed human intelligence by the year 2000. By 1970, however, they realized that such claims were too optimistic.

Although a few AI programs could demonstrate some level of machine intelligence in one or two toy problems, almost no AI projects could deal with a wider selection of tasks or more difficult real world problems.

The main difficulties for AI in the late sixties were:

- **Because AI researchers were developing general methods for broad classes of problems, early programs contained little or even no knowledge about a problem domain. To solve problems, programs applied a search strategy for trying out different combinations of small steps, until the right one was found. This method worked for toy problems, so it seemed reasonable that, if the programs could be scaled up to solve large problems, they would finally succeed. However this approach was wrong.**

  Easy problems can be solved in polynomial time, for a problem of size n, the time or number of steps needed to find the solution is a polynomial function of n.

  On the other hand, hard problems require times that are exponential functions of the problem size. While a polynomial-time algorithm is considered to be efficient, an exponential-time algorithm is inefficient, because its execution time increases rapidly with the problem size. The theory of NP-completeness developed in the early 1970-s,

showed the existence of a large class of non-deterministic polynomial problems that are NP-complete.
A problem is called NP-complete if its solution if one exists can be guessed and verified in polynomial time; non-deterministic means that no particular algorithm is followed to make the guess. The hardest problems in this class are NP-complete. Even with faster computers and larger memories, these problems are hard to solve

- Many of the problems that AI attempted to solve were too broad and too difficult. A typical task for early AI was machine translation. For example, the National Research Council , USA, funded the translation of Russian scientific papers after the launch of Sputnik in 1957. Initially, the project team tried simply replacing Russian words with English, using an electronic dictionary. However, it was soon found that translation requires a general understanding of the subject to choose the correct words. The task was too difficult. In 1966, all translation projects funded by the US government were canceled.

1.2.4 The technology of expert systems, or the key to success (early 1970s – mid 1980s)

The most important development in the 1970s was the realization that the problem domain for intelligent machines had to be sufficiently restricted.

The only way to deliver practical results was to solve typical cases in narrow areas of expertise by making large reasoning steps. In other words it became necessary to incorporate the human expertise into computer programs to make it perform at a human expert level. Such programs are called expert systems.

A 1986 survey reported a remarkable number of successful expert system applications in different areas: chemistry, electronics, engineering, geology, management, medicine, process control and military science.

However, in spite of a great number of successful developments and implementations of expert systems in different areas of human knowledge, it would be a mistake to overestimate the capability of this technology. The difficulties are rather complex and lie in both technical and sociological spheres. They include the following:

- Expert systems are restricted to a very narrow domain of expertise.
- Because of the narrow domain, expert systems are not as robust and flexible as a user might want.
- Expert systems have limited explanation capabilities. They can show the sequence of the rules they applied to reach a solution, but cannot relate accumulated heuristic knowledge to any deeper understanding of the problem domain.
- Expert systems are also difficult to verify and validate.
- Expert systems, especially the first generation, have little or no ability to learn from their experience.

1.2.5 How to make a machine learn, or the rebirth of neural networks (mid-1980s – onwards)

In the 1980s, because of the need for brain-like information processing, as well as the advances in computer technology and progress in neuroscience, the field of neural networks experienced a dramatic resurgence.

Artificial neural networks have come a long way from the early models to an interdisciplinary subject with roots in neuroscience, psychology, mathematics and engineering, and will continue to develop in both theory and practical applications.

1.2.6 Evolutionary computation, or learning by doing.

Natural intelligence is a product of evolution. Therefore, by simulating biological evolution, we might expect to discover how living systems are propelled towards high level intelligence. Nature learns by doing; biological systems are now told how to adapt to a specific environment – they simply compete for survival. The fittest species have a greater chance to reproduce, and thereby pass their genetic material to the next generation.

The evolutionary approach to AI is based on the computational models of natural selection and genetics. Evolutionary computation works by simulating a population of individuals, evaluating their performance, generating a new population, and repeating this process a number of times.

Evolutionary computation combines three main techniques: genetic algorithms, evolutionary strategies, and genetic programming.

Genetic programming represents an application of the genetic model of learning to programming. Its goal is to evolve not a coded representation of some problem, but rather a computer code that solves the problem. That is, genetic programming generates computer programs as the solution.

Genetic algorithms, evolutionary strategies and genetic programming represent rapidly growing areas of AI, and have great potential.

1.2.7 The new area of knowledge engineering, or computing with words (late 1980s – onwards)

Neural network technology offers more natural interaction with the real world than do systems based on symbolic reasoning. Neural networks can learn, adapt to changes in a problem's environment, establish patterns in situations where rules are not known, and deal with fuzzy or incomplete information. However, they lack explanation facilities and usually act as a black box. The process of training neural networks with current technologies is slow, and frequent retraining can cause serious difficulties.

Classic expert systems are especially good for closed-system applications with precise inputs and logical outputs. They use expert knowledge in the form of rules and, if required, can interact with the user to establish a particular fact. A major drawback is that human experts cannot always express their knowledge in terms of rules or explain the line of their reasoning. This can prevent the expert system from accumulating the necessary knowledge, and consequently lead to its failure. To overcome this limitation, neural computing can be used for extracting hidden knowledge in large data sets to obtain rules for expert systems.

Where acquired knowledge is incomplete, neural networks can refine the knowledge, and where the knowledge is inconsistent with some given data, neural networks can revise the rules.

Another very important technology dealing with vague, imprecise and uncertain knowledge and data is fuzzy logic. Most methods of handling imprecision in classic expert systems are based on the probability concept.

However experts do not usually think in probability values, but in such terms as often, generally, sometimes, occasionally and rarely. Fuzzy logic is concerned with the use of fuzzy values that capture the meaning of words, human reasoning and decision making.

At the heart of fuzzy logic lies the concept of a linguistic variable. The values of the linguistic variables are words rather than numbers. Similar to expert systems, fuzzy systems use IF –THEN rules to incorporate human knowledge, but these rules are fuzzy, such as:

IF speed is high THEN stopping_distance is long.

IF speed is low THEN stopping_distance is short.

<u>Where is knowledge engineering heading?</u>

Expert, neural and fuzzy systems have now matured and have been applied to a broad range of different problems, mainly in engineering, medicine, business, finances and management. Each technology handles the uncertainty and ambiguity of human knowledge differently, and each technology found its place in knowledge engineering.