Transactions on Information and Communications Technologies vol 19, © 1997 WIT Press, www.witpress.com, ISSN 1743-3517

The history of artificial intelligence or from the "Dark Ages" to the knowledge-based systems

M. Negnevitsky
Department of Electrical & Electronic Engineering
University of Tasmania
GPO Box 252-65 Hobart, Tasmania 7001, Australia
Email: Michael.Negnevitsky@eee.utas.edu.au

Abstract

We live in the era of the knowledge revolution when power of a nation is determined not by a number of soldiers in its army but the knowledge it possesses. Science, medicine engineering and business propel the nation towards a higher quality of life but also require highly qualified and skilful people. We are now adopting intelligent machines that can capture the expertise of such knowledgeable people and reason in a manner simular to humans. The desire for intelligent machines was just an elusive dream until the first computer was developed. This paper describes the history of artificial intelligence. The early computers were able to perform manipulation of large data bases effectively following prescribed algorithms but could not reason about the information provided. It gave a rise to the very question wether or not computers can think. Alan Turing proposed his imitation game (Turing test) and defined the intelligent behaviour of a computer as the ability to achieve the human-level performance in a cognitive task. Turing test provided a basis for the verification and validation of knowledge-based systems. In 1956, a summer workshop at Dartmouth College brought together ten researchers interested in the study of machine intelligence and a new science, artificial intelligence, was born. Expert, neural and fuzzy systems have now matured and applied to a broad range of different problems including engineering, medicine, economy, business and management. Each technology handles uncertainty and ambiguity of human knowledge differently, and each technology has found its place in knowledge engineering. A synergy of expert systems with fuzzy logic and neural computing improves adaptiveness, robustness, fault tolerance and speed of knowledge-based systems. Besides, computing with words makes them more "human".

Transactions on Information and Communications Technologies vol 19, © 1997 WIT Press, www.witpress.com, ISSN 1743-3517

1 The "Dark Ages" or the birth of artificial intelligence (1943 - 1956)

The first work recognised in the field of AI was presented by Warren McCulloch and Walter Pitts in 1943 (Culloch & Pitts, 1943). McCulloch majored in philosophy and also received his medical degree from Columbia University. As the Director of the Basic Research Laboratory in the Department of Psychiatry at the University of Illinois he conducted his research on central nervous system and made the first major contribution in AI, the model of neurons of the brain.

McCuloch and his co-author, a young mathematician Walter Pitts, proposed a model of artificial neural networks in which each neuron was postulated as being in binary state, that is, either in *on* or *off* condition. They demonstrated that the neural net model was, in fact, equivalent to the Turing machine, and proved that any computable function could be computed by some network of connected neurons. McCuloch and Pitts also showed that simple network structures could learn.

The neural net model stimulated both theoretical and experimental work to model the brain in the laboratory. However, experiments clearly demonstrated that the McCuloch and Pitts binary model of neurons was not correct. In fact, a neuron has highly non-linear characteristics and cannot be considered as a simple two-state device. In spite of this, McCuloch, the second after Alan Turing "founding father" of AI, has brought the corner stone of neural computing and artificial neural networks (ANN) and after the fall in the seventies, the field of ANN was reborn in the late eighties.

The third founder of AI was John von Neumann, the brilliant Hungarian-born mathematician. Since 1930 he joined the Princeton University lecturing in mathematical physics. He was a colleague of Allan Turing and they often visited each other. During World War II, von Neumann played one of the key roles in the Manhattan Project intended to build a nuclear bomb. He also became an adviser for the Electronic Numerical Integrator and Calculator project at the University of Pennsylvania and helped to design the Electronic Discrete Variable Calculator, a *stored program* machine. He was influenced by the McCuloch and Pitts ideas and their neural network model, and when in 1951, two graduate students in the Princeton mathematics department, Marvin Minsky and Dean Edmonds, build the first neural network computer, von Neumann greatly supported them.

One of the greatest patriarchs of the first generation of AI was Claude Shannon. He graduated from MIT and joined Bell Telephone Laboratories in 1941. Shannon shared Alan Turing's ideas on the possibility of machine intelligence. In 1950 he published a paper on chess-playing machines (Shannon, 1950). He pointed out that a typical chess game involved about 10^{120} possible moves. Even if the new Neumann-type computer could examine one move per microsecond it would take 10^{95} years to make the first move.

Thus he demonstrated the need to introduce heuristics to aid in the search of the solution.

Princeton University was also home to John McCarthy, another founder of AI. He convinced Martin Minsky and Claude Shannon to organise a summer workshop at Dartmouth College where McCarthy worked after graduation from Princeton. In 1956, they brought together researchers interested in the study of machine intelligence, artificial neural nets and automata theory. The workshop was sponsored by IBM. All together there were just ten researchers but this conference gave the birth of a new science called *artificial intelligence*. For the next twenty years the field of AI would be dominated by the participants of the Dartmouth two-month workshop and their students.

2 The rise of artificial intelligence or the era of great expectations (1956 - the late 1960s)

The early years of AI are characterised by the tremendous enthusiasm, great ideas and very limited success. Only a few years ago computers were introduced to perform just routine mathematical calculations, and now AI researchers were demonstrating that computers could do more than that. It was era of great expectations.

John McCarthy, one of the organisers of the Dartmouth workshop and the inventor of the term "artificial intelligence" moved from Dartmount to MIT. In 1958, he defined the high-level language *Lisp*. Lisp is the second-oldest (FORTRAN is just one year older) programming language, and is still in current use. Also in 1958, McCarthy presented his paper "Programs with Common Sense" (McCarthy, 1958). He proposed a program called the *Advice Taker* to search for solutions to general problems of the world. McCarthy demonstrated how his program could generate, for example, a plan to drive to the airport based on some simple axioms. Most importantly, the program was designed so that it could accept new axioms, or in other words new knowledge, in different areas of expertise without being reprogrammed. Thus the Advice Taker may be considered as the first complete hypothetical knowledge-based system incorporated the central principles of knowledge representation and reasoning.

Another organiser of the Dartmouth workshop, Marvin Minsky, also moved to MIT. However, unlike McCarthy with his tendency to formal logic, Minsky developed an anti-logical outlook on knowledge representation and reasoning. His theory of frames (Minsky, 1975*a*; Minsky, 1975*b*) was one of the major contributions in knowledge engineering.

Early works on neural computing and artificial neural networks started by McCulloch and Pitts were continued. Learning methods were improved and Frank Rosenblatt proved the *perceptron convergence theorem* demonstrating that his learning algorithm could adjust the connection strengths of a perceptron (Rosenblatt, 1962).

Transactions on Information and Communications Technologies vol 19, © 1997 WIT Press, www.witpress.com, ISSN 1743-3517

One of the most ambitious projects of the era of great expectations was the General Problem Solver (GPS) (Newell & Simon, 1961; Newell & Simon, 1972). Allen Newell and Herbert Simon from the Carnegie Mellon University developed a general-purpose program to simulate human-solving methods. GPS was probably the first attempt to separate the problem solving technique from the data. GPS was based on the technique referred now as *mean-end analysis*. Newell and Simon postulated that the problem to be solved could be defined in terms of *states*. The mean-end analysis was used to determine a difference between the current state and the desirable state or the *goal state* of the problem, and to choose and apply appropriate *operators* to reach the goal state. If the goal state could not be immediately reached from the current state the 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 program was based on the formal logic and therefore it could generate an infinite number of possible operators. In other words, GPS was inherently inefficient. Finally it was realised that due to computer time and memory requirements it would be impossible to use GPS for solving real-world problems.

In summary, we can say that in the sixties, AI researchers attempted to simulate the complex thinking process by inventing *general methods* for solving *broad classes of problems*. They used the general-purpose search mechanism to find a solution to the problem. Such approaches, now referred as *weak methods*, applied weak information about the problem domain. It led to weak performance of the programs developed.

However, it was also time when the field of AI attracted great scientists who introduced new fundamental ideas in such areas as knowledge representation, learning algorithms, neural computing and computing with words. These ideas could not be implemented yet because of limited capabilities of the computers, but two decades later they have led to the development of real-life practical applications.

It may be interesting to note that Lotfi Zadeh, professor from the University of California at Berkeley, published his famous paper "Fuzzy Sets" also in sixties, in 1965 (Zadeh, 1965). This paper now is considered as a foundation of the fuzzy set theory. Two decades later fuzzy researchers have built hundreds of smart machines and intelligent systems.

However, by 1970, the euphoria surrounding AI was gone and most government funding for AI projects were also cancelled. AI was still a relatively new field, academic in nature with very few practical applications mostly for playing games. As examples, we could mention Arthur Samuel's checkers program (Samuel, 1959; Samuel, 1967) and the MacHack 6 chess program (Greenblatt et al., 1967), the first chess program competed successfully with humans. Meanwhile, to the outsider, the achieved results would be seen as toys, no AI system at that time could manage real-world problems.

Transactions on Information and Communications Technologies vol 19, © 1997 WIT Press, www.witpress.com, ISSN 1743-3517

3 Unfulfilled promises or an impact of reality (the late 1960s - the early 1970s)

Since the mid-fifties, 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, it was finally understood that such claims were too optimistic. Although a few AI programs could demonstrate some level of machine intelligence on one or two simple toy problems, almost all AI projects failed to deal with wider selection of tasks or with more difficult real-world problems.

Let us now identify the most significant difficulties which led to the fall of AI in the beginning of seventies.

- AI researchers were developing general methods for broad classes of problems and therefore early programs contained little or even no knowledge about a problem domain. They applied the solution search strategy trying out different combinations of small steps until the right one was found. This approach was quite feasible for simple toy problems and scientists believed that by "scaling up" to large problems they could finally succeed. However, many of important practical problems are logically intractable. The theory of NP-completeness (Cook, 1971; Karp, 1972) developed in the early seventies showed the existence of a large class of non-deterministic polynomial problems, NP problems, that are NP-complete, i.e. likely to be unmanageable. A problem belongs to NP class if there is some algorithm that can guess a solution and then verify it in polynomial time. The hardest problems in this class are NP-complete. These problems cannot be solved using combinatorial search and reasoning techniques, and thus even the application of faster computers and larger memories could not lead to anywhere.
- 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 the first artificial satellite (Sputnik) in 1957. Initially, the attempt was made simply to replace Russian words by English using an electronic dictionary. However, it was soon found that translation requires general understanding of the subject in order to define the correct meaning of the words. This task was too difficult and, thus the project failed to achieve any practical results in translation of a general scientific text. As a response, in 1966 all translation projects funded by the US government were cancelled.

In 1971, as a result of the Lighthill report (Lighthill, 1973), the British government also suspended support for AI research. Sir James Lighthill was commissioned by the Science Research Council of Great Britain to review a current state of AI. He did not found any major or even significant impact that was promised by AI researchers and, therefore, he saw no need for the very existence of a separate science called artificial intelligence.

4 The technology of expert systems or the key to success (the early 1970s - the mid-1980s)

Probably the most important discovery made in the seventies was understanding that the domain for intelligent machines had to be sufficiently restricted. Previously AI researchers believed that clever search algorithms and reasoning techniques could be invented to emulate general human-like problem solving methods. A general-purpose search mechanism relied upon elementary reasoning steps to find complete solutions and used week knowledge about domain. When weak methods failed, it was finally realised that the only way to deliver practical results was to solve typical cases in narrow areas of expertise making large reasoning steps. In other words, it was generally accepted that to make a machine to solve a problem, one have to know the solution already.

The DENDRAL program can be considered as a typical example of the emerging technology (Buchanan et al., 1969). DENDRAL was developed at Stanford University to perform chemical analysis. The project was highly supported by NASA because an unmanned spacecraft had to be launched on Mars and a program was required to determine the molecular structure of the Marsian soil based on the mass spectral data provided by a mass spectrometer. Edward Feigenbaum, a formal student of Herbert Simon, Bruce Buchanan, a computer scientist, and Joshua Lederberg, a Nobel prize winner in genetics, formed a team to solve this challenging problem. The traditional method of solving such problems relies on a generate-and-test technique. All possible molecular structures consistent with the mass spectrogram are generated first. Then the mass spectrum is determined or predicted for each structure and tested against the actual spectrum. However, this method failed because millions of possible structures could be generated. The problem rapidly became intractable even for decent-sized molecules. To add to the difficulties of this challenge, there was no scientific algorithm for mapping the mass spectrum into its molecular structure. However, analytical chemists, such as Lederberg, could solve this problem using their skills, experience and expertise. They could enormously reduce the number of possible structures by looking for well known patterns of peaks in the spectrum, and thus provide just a few feasible solutions for the further examination. Therefore, the job of Feigenbaum was to incorporate the expertise of Lederberg into a computer program to make it to perform at a human expert level. Such programs later were called *expert systems*. To understand and adopt Lederberg knowledge and operate with his terminology, Feigenbaum had to learn basic ideas in chemistry and spectral analysis. However, it became apparent that Feigenbaum used not only rules of chemistry but also his own heuristics or rules-of-thumb based on his experience and even guessing. Soon Feigenbaum identified one of the major difficulties in the project termed him as the "knowledge acquisition bottleneck" - how to extract knowledge from human experts to apply it in computers. To articulate his knowledge, Lederberg even needed to study basics in computing. Working as a team, Feigenbaum, Buchanan and Lederberg developed DENDRAL, the first successful knowledge-based system. The DENDRAL team pointed out the key to success: all the relevant theoretical knowledge in the domain was mapped over from its general form to highly specific form ("cookbook recipes" in the form of rules) (Feigenbaum et al., 1971).

The significance of DENDRAL can be summarised as follows:

- DENDRAL marked a major "paradigm shift" in AI - a shift from general-purpose, knowledge-sparse weak methods to domain-specific, knowledge-intensive techniques.
- The aim of the project was to develop a computer program to attain the kind of performance normally reserved for experienced human chemists. Using heuristics in the form of high-quality specific rules, rules-of-thumb, elicited from human experts, the DENDRAL team proved that computers could play the role of experts in some narrow problem area.
- The DENDRAL project originated the fundamental idea of the new methodology of expert systems - *knowledge engineering*, techniques of capturing, analysing and expressing into rules an expert's "know-how".

DENDRAL proved to be a useful analytical tool for chemists and was marketed commercially in the United States.

The next major project undertaken by Feigenbaum and others at Stanford University was in the area of medical diagnosis. The project called MYCIN started in 1972 and later became the Ph.D thesis of Edward Shortliffe published in 1976 (Shortliffe, 1976). MYCIN was a rule based expert system for the diagnosis of infectious blood diseases. It also provided a doctor with therapeutic advice in a convenient user-friendly manner.

MYCIN shared a number of characteristic common for early expert systems, including:

- MYCIN was able to perform at a level equivalent to human experts in the field and considerably better than junior doctors.
- The MYCIN knowledge comprised approximately 450 independent rules of IF-THEN form was derived from hand-crafted knowledge in a narrow domain through extensive interviewing of experts.
- The clean separation of the knowledge incorporated in the form of rules and the reasoning mechanism. The system developer could easily manipulate knowledge in the system by inserting or deleting some rules. Moreover, old rules could be removed completely. For example, a domain-independent version of MYCIN called EMYCIN (Empty MYCIN) was later produced at Stanford University (van Melle, 1979; van Melle et al., 1981). It had all the features of the MYCIN system except the knowledge

of infectious blood diseases. EMYCIN facilitated the development of a variety of diagnostic applications. System developers had to just add new knowledge in the form of rules to obtain a new application.

MYCIN also introduced a few new features. Rules incorporated in MYCIN reflected the uncertainty associated with knowledge, in this case with medical diagnosis. It tested a rule conditions, IF part, against available data or data requested from the physician. When appropriate, MYCIN inferred the truth of a condition using a calculus of uncertainty called *certainty factors*. Reasoning in the face of uncertainty was the most important part of the system.

Another probabilistic system generated enormous publicity was PROSPECTOR, an expert system for mineral exploration (Duda et al., 1979). PROSPECTOR was developed by the Stanford Research Institute. The project started in 1974 and continued until 1983. Nine experts contributed their knowledge and expertise. To represent their knowledge PROSPECTOR used a combined structure incorporated rules and semantic network. PROSPECTOR was a large expert system with over 1000 rules which represented extensive domain knowledge. It also had a sophisticated support package, knowledge acquisition system (KAS) to facilitate the acquisition of knowledge in the system.

PROSPECTOR operates as follows. The user, an exploration geologist, is asked to input the characteristics of a suspected deposit: the geological setting, structural controls, kinds of rocks and minerals. Then the program compares these characteristics with models of ore deposits and, if necessary, engages the user in a dialogue to obtain additional relevant information. Finally PROSPECTOR makes an assessment of the suspected mineral deposit and present its conclusion. The system can also explain he steps used to reach the conclusion.

In exploration geology, important decisions are usually made in the face of uncertainty when knowledge is incomplete or fuzzy. To deal with such knowledge, PROSPECTOR incorporated Bayes' rules of evidence to propagate uncertainties through he system. PROSPECTOR performed at the level of an expert geologist and proved itself at searching for new mineral exploration deposits. In 1980, it predicted a molybdenum deposit near Mount Tolman in eastern Washington. Subsequent drilling by a mining company confirmed the prediction as having a value of over $100 million. One could hardly ever expect a better justification of the expert system use.

Expert systems mentioned above have become classical now. A growing number of successful applications of expert systems in the late seventies indicated that AI technology moved from the research laboratories to the commercial environment. During this period, however, most expert systems were developed using special AI languages such as LISP, PROLOG and OPS based on powerful workstations. As a result of a need to have rather expensive hardware and to use rather complicated programming languages, the challenge of the expert system development was left in the hands of a very few research groups originally located in the Stanford University, MIT, Stanford Research

Institute and Carnegue-Mellon University. Only in the eighties, when personal computers, PCs, appeared and then saturated the market, and easy-to-use expert system development tools, shells, were introduced, the opportunity to develop an expert system passed in the hands of ordinary researchers and engineers in all disciplines.

A survey published by Donald Waterman (Waterman, 1986) in 1986 revealed remarkable number of successful expert systems applications in different areas. Waterman found and described nearly 200 expert systems used in chemistry, electronics, engineering, geology, management, medicine, process control and military science. The majority of the applications were in the field of medical diagnosis. Seven years later a similar survey was conducted by John Durkin (Durkin, 1994). His survey included reports on over 2500 developed expert systems and indicated the growing activity in such areas as business and manufacturing. These areas accounted for approximately 60% of the applications. This clearly demonstrates the maturity of the expert system technology.

Are expert systems really the key to success in any field? 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. Difficulties are rather complex and lie in both technical and sociological spheres. They include following:

- Expert systems are restricted to a very narrow domain of expertise. For example, MYCIN developed for the diagnosis infectious blood diseases in fact lacks any real knowledge of human physiology, and if a patient has more than one disease, we cannot rely on MYCIN because providing a therapy only for the most probable infectious blood disease is not longer enough.
- Because of the narrow domain, expert systems are not as robust and flexible as a user might want. Furthermore, expert systems could have difficulties in recognising domain boundaries. When given a task different from the typical problems, an expert system might attempt to solve it and fail as a result in rather unpredictable ways.
- Expert systems have limited explanation capabilities. They are able to show a sequence of the rules 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. No general technique has been developed yet for analysing the system completeness and consistency. Heuristic rules represent knowledge in abstract form and lack even basic understanding of the domain area. It makes a task of the identification of incorrect, incomplete or inconsistent knowledge very difficult.
- Expert systems, especially the first generation, have little or no learning capabilities from their experience. Expert systems are handcrafted and cannot be developed quickly. According to Waterman (Waterman, 1986),

it takes from five to ten person-years to build an expert system to solve a moderately difficult problem. Complex systems such as DENDRAL, MYCIN or PROSPECTOR can demand over 30 person years to complete. This large effort, however, might be difficult to justify because any improvement of the expert system performance would depend on the further attention from its developers. Moreover, it would cause doubts about the intelligence of such systems.

Despite of all these difficulties, expert systems have made the breakthrough and proved their value in a number of important applications.

5 How to make a machine to learn or the rebirth of neural networks (the mid-1980s - present)

In the mid-eighties, researchers, engineers and experts finally found that building an expert system requires much more than just buying a reasoning system or expert system shell and putting enough rules in it. Disillusions concerning the applicability of the expert system technology even led to a prediction of an AI "winter" with severely squeezed funding of AI projects. This caused AI researchers to have a new look on the side of neural networks.

By the late sixties, according to Cowan (Cowan, 1990), most of the basic ideas and concepts necessary for neural computing were already formulated. However, we had to wait until the mid-eighties for the solutions to emerge. The major reason was technological. There were no PCs or powerful workstations to model and carry out experiments with artificial neural nets. The other reasons include psychological and financial barriers. For example, in 1969, Minsky and Papert in the book *Perceptrons* (Minsky & Papert, 1969) mathematically demonstrated fundamental limitations of computational capabilities of one-layer perceptrons. They also stated that there was no reason to expect that more complex multilayer perceptrons could represent much, and thus certainly did not encourage anyone to work on perceptrons. As a result, majority of AI researchers deserted the field of ANN in the seventies. However, psychologies and neuro scientists continued their important work.

In the eighties, due to the need for brainlike information processing, advances in computer technology and progress in neuroscience the field of neural networks experienced a dramatic resurgence. Major contributions to both theory and design were made on several fronts. Grossberg established a new principle of self-organisation which provided the basis of a new class of neural networks based on the *adaptive resonance theory* (Grossberg, 1980). Hopfield introduced a new class of neural networks with feedback, *Hopfield networks*, attracted a great attention in the eighties (Hopfield, 1982). In 1982, Kohonen published a paper on *self-organising maps* (Kohonen, 1982). Another important paper written by Barto, Sutton and Anderson appeared in 1983 (Barto, 1983). This work was on *reinforcement learning* and its application in control. But the real breakthrough came in 1986 when the *back-propagation learning algorithm* first introduced by Bryson and Ho in 1969 (Bryson & Ho, 1969) was reinvented by Rumelhart and McClelland in the two-volume book, *Parallel Distributed Processing: Explorations in the Microstructures of Cognition* (Rumelhart & McClelland, 1986). In the same time, back-propagation learning was also discovered by Parker (Parker, 1987) and LeCun (LeCun, 1988) and since then has become the most popular technique used for the training of multilayer perceptrons. In 1988, Broomhead and Lowe (Broomhead & Lowe, 1988) found a procedure to design *layered feedforward networks* using radial basic functions, an alternative to multilayer perceptrons.

Artificial neural networks have covered a long way from the early basic models of McCulloch and Pitts to an interdisciplinary subject with roots in the neuroscience, psychology, mathematics and engineering, and will continue their development in both theory and practical applications. In conclusion, however, it is fair to highlight just two publications, the 1982 paper by Hopfield (Hopfield, 1982) and the 1986 two-volume book by Rumelhart and McClelland (Rumelhart & McClelland, 1986) as the most significant and influential work responsible for the rebirth of neural networks in eighties.

6 The new era of knowledge engineering or computing with words (the late 1980s - present)

The neural network technology offers more natural interaction with the real world than systems based on symbolic reasoning. Neural nets provide such an important human characteristic of problem solving as learning, adaptive to changes in the problem environment, can process a large amount of data to establish patterns in situations where rules are not known, and can deal with fuzzy or incomplete information. However, ANNs lack explanation facilities and usually act as a black box. A process of the neural network training with current technologies takes an excessive time, and thus, the need for frequent retraining can cause serious difficulties.

Although in some special cases, particularly in knowledge-poor situations, ANNs can solve problems better than expert systems, the two technologies are not in competition now. They rather nicely complement each other.

Classical 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 of the expert system approach occurs as a result that human experts do not always can express their knowledge in terms of rules, explain the line of their reasoning and, moreover, they may explain it incorrectly. In many cases this can prevent to accumulate the necessary knowledge in the expert system, and consequently may lead to the failure of the expert system technology. To overcome this limitation, neural computing can be used for extracting hidden knowledge in large data sets to obtain rules for expert systems (Medsker & Leibowitz, 1994; Zahedi, 1993). ANNs can also be used for changing incorrect rules in traditional rule-based expert systems (Omlin & Giles, 1996). In cases where acquired knowledge is incomplete, networks perform knowledge refinement, and in cases where the prior knowledge is inconsistent with some given data, networks perform rule revision.

Another very important technology dealing with vague, imprecise and uncertain knowledge and data is *fuzzy logic*. Most methods of handling imprecision in classical expert systems are based on the probability concept. MYCIN, for example, introduced certainty factors and PROSPECTOR incorporated Bayes' rules to propagate uncertainties. However, experts do not usually think in probability values, but in such terms as often, generally, sometimes, occasionally, rarely, etc. Fuzzy logic is concerned with the use of fuzzy values which capture the meaning of words, human reasoning and decision making. As a method to encode and apply human knowledge in a form that accurately reflects an expert understanding of difficult, complex problems, fuzzy logic provides the way to break through the computational bottlenecks of traditional expert systems.

In the heart of fuzzy logic lies the concept of a linguistic variable. Values of the linguistic variable 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 braking_distance is long

IF speed is low THEN braking_distance is short.

Fuzzy logic or *fuzzy set theory* was first introduces by professor Lotfi Zadeh, Berkeley's electrical engineering department chairman, in 1965 (Zadeh, 1965). It provided a means of computing with words. However, acceptance of fuzzy set theory by technical community was slow and difficult. Part of the problem was the provocative name - "fuzzy" and some people rejected the theory outright because of its name, even without knowing the content. Eventually, fuzzy theory ignored on the West was taken seriously on the East - by the Japanese. Fuzzy logic has been successfully used in numerous consumer products since 1987. Now Japanese consumer products such as dishwashers, washing machines, air conditioners, television sets, copiers and even cars apply fuzzy technology. The introduction of fuzzy products gave rise to tremendous interest in this apparently "new" technology developed over 30 years ago. Hundreds of books have been published on this topic and over 15000 technical papers have been written. We can mention here just a few books that have already become classical, *Fuzzy Sets, Neural Networks and Soft Computing* edited by Ronald Yager and Lotfi Zadeh (Yager & Zadeh, 1994), *The Fuzzy Systems Handbook* by Earl Cox (Cox, 1994), *Neural Networks and Fuzzy Systems* by Bart Kosko (Kosko, 1992), *Expert Systems and Fuzzy Systems* by Constantin Negoita (Negoita, 1985) and also a best-seller science book, *Fuzzy*

Thinking by Bart Kosko (Kosko, 1993) which did a lot to popularise the field of fuzzy logic in last few years.

Most of fuzzy logic applications have been in the area of process and control engineering. However, control fuzzy systems utilise only a small part of the knowledge representation power of fuzzy logic. Benefits derived from the application of fuzzy logic models in knowledge-based and decision support systems can be summarised as (Cox, 1994; Turban, 1995):

- *Improved computational power:* Fuzzy rule-based systems perform faster than conventional expert systems and require fewer rules. A fuzzy expert system merges the rules making them more powerful. Lotfi Zadeh believes that in a few years most expert systems will use fuzzy logic and it will make them capable to solve highly nonlinear and computationally difficult problems.
- *Improved cognitive modelling:* Fuzzy systems allow to encode knowledge in a form which directly reflects expert thinking about a complex problem. Experts usually think in imprecise terms such as *high* and *low*, *fast* and *slow*, *heavy* and *light*, and etc. Moreover, they also use terms like *very often* and *almost never*, *usually* and *hardly ever*, *frequently* and *occasionally*. In order to build conventional rules, we need to define the crisp boundaries for these terms, and thus, to break down the expertise into fragments. However, this fragmentation leads to the poor performance of conventional expert systems when they deal with highly complex problems. Fuzzy expert systems model imprecise information, capture expertise much closer to the way it is represented in the expert mind, and thus, improve cognitive modelling of the problem.
- *The ability to represent multiple experts:* Conventional expert systems are built for very narrow domain with clearly defined expertise. It makes the system performance fully dependent on the right choice of experts. The identification of experts is a complicated task in the real-world environment, and a commonly used strategy is to find just one expert. However, when more complex knowledge-based systems are being built or when expertise is not well defined, there could be a need for *multiple experts*. Benefits of the use of multiple experts include expanded domain, synthesis of expertise, and eliminating the need for a world-class expert who is very expensive and difficult to access. However, it is seldom the case when all the experts can find at least a close agreement. There are often differences in opinions and even conflicts. This is especially true when knowledge-based systems are to be developed for business and management where no simple solution exists and conflicting views should be taken into account. Fuzzy expert systems can help to represent expertise of multiple experts even with directly opposite opinions.

Although fuzzy systems allow to express expert knowledge in more natural way, they still depend on the rules extracted from the experts, and thus might be smart or dumb. Some experts can provide very clever fuzzy rules, but some just guess and even may get them wrong. Therefore, all rules must be tested and tuned. This can be a prolonged and tedious process. For example, Hitachi engineers tested and tuned fuzzy rules to guide the Sendal Subway System for a few years. But they used only *54 rules*!

Using fuzzy logic development tools we can easily build a simple fuzzy system but then we may spend days, weeks and even months trying out new rules and tuning our system. How to make this process faster or , in other words, how to generate good fuzzy rules automatically?

In recent years, several methods based on the neural network technology have been employed to search numerical data for fuzzy rules. Adaptive or neural fuzzy systems can find new fuzzy rules, change or tune existing ones based on the data provided. In other words, data in - rules out, or experience in - common sense out.

So, where is knowledge engineering heading for?

Expert, neural and fuzzy systems have now matured and applied to a broad range of different problems including engineering, medicine, economy, business and management. Each technology handles uncertainty and ambiguity of human knowledge differently, and each technology has found its place in knowledge engineering. They do not compete any more, but rather complement each other. A synergy of expert systems with fuzzy logic and neural computing improves adaptiveness, robustness, fault tolerance and speed of knowledge-based systems. Besides, computing with words makes them more "human". It is now a commonly accepted practice to build intelligent systems using existing theories rather than to propose new ones, and to apply these systems to real-world problems rather than to "toy" problems.

7 Summary

We live in the era of the knowledge revolution when power of a nation is determined not by a number of soldiers in its army but the knowledge it possesses. Science, medicine engineering and business propel the nation towards a higher quality of life but also require highly qualified and skilful people. We are now adopting intelligent machines that can capture the expertise of such knowledgeable people and reason in a manner simular to humans.

The desire for intelligent machines was just an elusive dream until the first computer was developed. The early computers were able to perform manipulation of large data bases effectively following prescribed algorithms but could not reason about the information provided. It gave a rise to the very question wether or not computers can think. Alan Turing proposed his imitation game (Turing test) and defined the intelligent behaviour of a computer as the ability to achieve the human-level performance in a cognitive task.

Turing test provided a basis for the verification and validation of knowledge-based systems.

In 1956, a summer workshop at Dartmouth College brought together ten researchers interested in the study of machine intelligence and a new science, artificial intelligence, was born.

Since the early fifties an AI technology have covered a way from a curiosity of a few researchers to a valuable tool to support humans in their decision making. We have seen historical cycles of AI from the era of great ideas and great expectations in the sixties to the fall and cutbacks in funding in the early seventies, from the development of the first expert systems such as DENDRAL, MYCIN and PROSPECTOR in the seventies to the maturity of the expert system technology and its massive applications in different areas in the eighties and nighties, from a simple binary model of neurons proposed in forties to a dramatic resurgence of the field of artificial neural networks in the eighties, from the introduction of fuzzy set theory and its ignorance on the West in sixties and seventies to numerous "fuzzy" consumer products offered by the Japanese in the eighties and computing with words put forward in the nighties.

The development of expert systems brought to light the new art and science called knowledge engineering. Knowledge engineering can be now defined as the process of building intelligent systems. Today it deals not only with expert systems but also with neural networks and fuzzy logic. Knowledge engineering is still an art rather than engineering but attempts have been already made to extract rules automatically from numerical data using the neural network technology.

Table provides a brief summary of the key events in the history of AI and knowledge engineering from the first work in the field of AI published by McCulloch and Pitts in 1943 to recent events reflecting a tendency to combine strengths of expert systems, fuzzy logic and neural computing in modern knowledge-based systems capable to compute with words.

The most important lessons learned in this chapter are:

- Intelligence is the ability to learn and understand, to solve problems and to make decisions.
- Artificial intelligence is a science that has established its goal in making machines to do things that would require intelligence if done by men.
- A machine is thought intelligent if it can achieve the human-level performance in some cognitive task. To build an intelligent machine, we need to capture, organise and use a human expert knowledge in some problem area.
- An understanding that the domain for intelligent machines had to be sufficiently restricted marked a major "paradigm shift" in AI from general-purpose, knowledge-sparse week methods to domain-specific, knowledge-intensive techniques. This led to the development of expert systems, computer programs capable to perform at a human expert level in a narrow problem area. Expert systems use human knowledge and expertise in the form of specific rules, and are distinguished by the clean separation of the

knowledge and the reasoning mechanism. They can also explain their reasoning procedures.

- One of the major difficulties in the process of building intelligent machines, or in other words in knowledge engineering, is the "knowledge acquisition bottleneck" - extracting knowledge from human experts.
- Experts think in imprecise terms such as very often and almost never, usually and hardly ever, frequently and occasionally and use linguistic variables such as high and low, fast and slow, heavy and light. Fuzzy logic or fuzzy set theory provides a means to compute with words. It concentrates on the use of fuzzy values which capture the meaning of words, human reasoning and decision making process and provides the way to break through the computational burden of traditional expert systems.
- Expert systems have little or no learning capabilities or self improvement through experience. They are "hand crafted" and demand large efforts for their development. It can take from five to ten person-years to build even a moderate expert system. Machine learning can help to accelerate this process significantly and enhance quality of the knowledge by adding new rules or changing incorrect ones.
- Artificial neural networks inspired by biological neural networks learn from historical cases and make possible to generate rules automatically and thus avoid the tedious and expensive processes of knowledge acquisition, validation and revision.
- Integrations of expert systems and ANNs, and fuzzy logic and ANNs improve adaptiveness, fault tolerance and speed of knowledge-based systems.

Table: An overview of the key events in the history of AI and knowledge engineering.

Period	***Key Events***
The birth of Artificial Intelligence (1943 - 1956)	McCulloch and Pitts, *A Logical Calculus of the Ideas Immanent in Nervous Activity*, 1943 Alan Turing, *Computing Machinery and Intelligence*, 1950 The Electronic Numerical Integrator and Calculator project (John von Neumann) Claude Shannon, *Programming a Computer for Playing Chess*, 1950 The Dartmouth College summer workshop on machine intelligence, artificial neural nets and automata theory, 1956

Table *Continued*

Period	***Key Events***
The rise of Artificial Intelligence (1956 - the late 1960s)	*LISP* (John McCarthy) The General Problem Solver (GPR) project (Newell and Simon) Newell and Simon, *Human Problem Solving*, 1972 Marvin Minsky, *A Framework for Representing Knowledge*, 1975
The fall of Artificial Intelligence (the late 1960s-the early 1970s)	Cook, *The Complexity of Theorem Proving Procedures*, 1971 Karp, *Reducibility Among Combinatorial Problems*, 1972 The Lighthill report, 1971
The discovery of Expert Systems (the early 1970s - the mid-1980s)	DENDRAL (Feigenbaum, Buchaman and Leberberg, Stanford University) MYCIN (Feigenbaum and Shortliffe, Stanford University) PROSPECTOR (The Stanford Research Institute) PROLOG - *a logic programming language* (Colmerauer, Roussel and Kowalski, France) EMYCIN (Stanford University) Donald Waterman, *A Guide to Expert Systems*, 1986
The rebirth of Artificial Neural Networks (1965 - present)	Hopfield, *Neural Networks and Physical Systems with Emergent Collective Computational Abilities*, 1982 Kohonen, *Self-Organized Formation of Topologically Correct Feature Maps*, 1982 Rumelhart and McClelland, *Parallel Distributed Processing*, 1986 The First IEEE International Conference on Neural Networks, 1987 Simon Haykin, *Neural Networks*, 1994 Neural Network, MATLAB Application Toolbox (The MathWork, Inc.)

Table *Continued*

Period	*Key Events*
Computing with words (the 1980s - present)	Lotfi Zadeh, *Fuzzy Sets*, 1965 Lotfi Zadeh, *Fuzzy Algorithms*, 1969 Mandani, *Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis*, 1977 Sugeno, *Fuzzy Theory*, 1983 Japanese "fuzzy" consumer products (dishwashers, washing machines, air conditioners, television sets, copiers) Sendal Subway System (Hitachi, Japan), 1986 Constantin Negoita, *Expert Systems and Fuzzy Systems*, 1985 The First IEEE International Conference on Fuzzy Systems, 1992 Bart Kosko, *Neural Networks and Fuzzy Systems*, 1992 Bart Kosko, *Fuzzy Thinking*, 1993 Ronald Yager and Lotfi Zadeh, *Fuzzy Sets, Neural Networks and Soft Computing*, 1994 Earl Cox, *The Fuzzy Systems Handbook*, 1994 Bart Kosko, *Fuzzy Engineering*, 1996 Lotfi Zadeh, *Computing with Words - A Paradigm Shift*, 1996 Fuzzy Logic, MATLAB Application Toolbox (The MathWork, Inc.)

8 References

Barto, A.G., Sutton, R.S. and Anderson C.W. (1983). Neurolike Adaptive Elements that Can Solve Difficult Learning Control Problems, *IEEE Transactions on Systems, Man and Cybernetics*, SMC-13, pp. 834-846.

Boden, M.A. (1977). *Artificial Intelligence and Natural Man*, Basic Books, New York.

Broomhead, D.S. and Lowe, D. (1988). Multivariable Functional Interpolation and Adaptive Networks, Complex Systems, 2, pp. 321-355.

Bryson, A.E. and Ho, Y.-C. (1969). *Applied Optimal Control*. Blaisdell, New York.

Buchanan, B.G., Sutherland, G.L., and Feigenbaum, E.A. (1969). Heuristic DENDRAL: a Program for Generating Explanatory Hypotheses in Organic Chemistry. *Machine Intelligence 4*, B. Meltzer, D. Michie and M. Swann, eds., Edinburgh University Press, Edinburgh, Scotland, pp. 209-254.

Cook, S.A. (1971). The Complexity of Theorem Proving Procedures. *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, New York, pp. 151-158.

Cowan, J.D. (1990). Neural Networks: The Early Days. *Advances in Neural Information Processing Systems 2*, D.S. Tourefzky, ed., San Mateo, CA: Morgan Kaufman, pp. 828-842.

Cox, E. (1994). *The Fuzzy Systems Handbook: A Practitioner's Guide to Building, Using, and Maintaining Fuzzy Systems*, Academic Press, Cambridge, 1994.

Duda, R., Gaschnig, J. and Hart, P. (1979). Model Design in the PROSPECTOR Consultant System for Mineral Exploration. *Expert Systems in the Microelectronic Age*, D. Michie, ed., Edinburgh University Press, Edinburgh, Scotland, pp. 153-167.

Durkin, J. (1994). *Expert Systems Design and Development*, Printice Hall, Englewood Cliffs, New Jersey.

Feigenbaum, E.A., Buchanan, B.G. and Lederberg, J. (1971). On Generality and Problem Solving: A Case Study Using the DENDRAL Program. *Machine Intelligence 6*, B.Meltzer and D. Michie, eds., Edinburgh University Press, Edinburgh, Scotland, pp. 165-190.

Greenblatt, R.D., Eastlake, D.E. and Crocker, S.D. (1967). The Greenblatt Chess Program. *Proceedings of the Fall Joint Computer Conference*, pp. 801-810.

Grossberg, S. (1980). How does a Brain Build a Cognitive Code?, *Psychological Review*, 87, pp. 1-51.

Hopfield, J.J. (1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities, *Proceedings of the National Academy of Sciences of the U.S.A.*, 79, pp. 2554-2558.

Karp, R.M. (1972). Reducibility Among Combinatorial Problems. *Complexity of Computer Computations*. R.E. Miller and J.W. Thatcher, eds., Plenum, New York, pp. 85-103.

Kohonen, T. (1982). Self-Organized Formation of Topologically Correct Feature Maps, *Biological Cybernetics*, 43, pp. 59-69.

Transactions on Information and Communications Technologies vol 19, © 1997 WIT Press, www.witpress.com, ISSN 1743-3517

Kosko, B. (1992). *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice-Hall, Englewood Cliffs, NJ.

Kosko, B. (1993). *Fuzzy Thinking: The New Science of Fuzzy Logic*, Hyperion, New York.

LeCun, Y. (1988). A Theoretical Framework for Back-Propagation, *Proceedings of the 1988 Connectionist Models Summer School*, D.Touretzky, G.Hilton and T.Sejnowski, eds., San Mateo, CA: Morgan Kaufmann, pp. 21-28.

Lighthill, J. (1973). Artificial Intelligence: A General Survey. *Artificial Intelligence: A Paper Symposium*. J. Lighthill, N.S. Sutherland, R.M. Needham, H.C. Longuest-Higgins and D. Michie, eds., Science Research Council of Great Britain, London.

McCarthy, J. (1958). Programs With Common Sense. *Proceedings of the Symposium on Mechanisation of Thought Processes*, Vol. 1, London, pp.77-84.

McCulloch, W.S. and Pitts, W. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115-137.

Medsker, L. and Leibowitz, J. (1994). *Design and Development of Expert Systems and Neural Computing*. New York: Macmillan.

Minsky, M.L. and Papert, S.A. (1969). *Perceptrons*, Cambridge, MA: MIT Press.

Minsky, M.L. (1975*a*). A Framework for Representing Knowledge: *The Psychology of Computer Vision*, P.H.Winston, ed., McGraw-Hill, New York, , pp. 211-277.

Minsky, M.L. (1975*b*). Frame System Theory, in Thinking: *Reading in Cognitive Science*, P.N. Johnson-Laird and P.C.Watson, eds., University Press, Cambridge.

Negoita, C. V. (1985). *Expert Systems and Fuzzy Systems*, The Benjamin/Cummings Publishing Company, Menlo Park, California.

Newell, A. and Simon, H.A. (1961). GPS, a Program that Simulates Human Thought: *Lernende Automatten*. H.Billing, ed., R.Oldenbourg, Munich, Germany, pp. 109-124.

Newell, A. and Simon, H.A. (1972). *Human Problem Solving*. Printice-Hall, Englewood Cliffs, N.J.

Omlin, C.W. and Giles, C.L. (1996). Rule Revision with Recurrent Neural Networks, *IEEE Transactions on Knowledge and Data Engineering*, Vol.8, No.1, pp. 183-188.

Parker, D.B. (1987). Optimal Algorithms for Adaptive Networks: Second Order Back Propagation, Second Order Direct propagation, and Second Order Hebbian Learning, *Proceedings of the IEEE 1st International Conference on Neural Networks*, San Diego, CA, Vol.2, pp. 593-600.

Rosenblatt, F. (1962). *Principles of Neurodynamics*. Spartan, Chicago.

Rumelhart, D.E. and McClelland, J.L., eds. (1986). *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, MA: MIT Press, Cambridge, Massachusetts. In two volumes.

Samuel, A.L. (1959). Some Studies in Machine learning Using the Game of Checkers. *IBM Journal of Research and Development*, Vol.3, No.3, pp.210-229.

Samuel, A.L. (1967). Some Studies in Machine learning Using the Game of Checkers II - Recent Progress. *IBM Journal of Research and Development*, Vol.11, No.6, pp. 601-617.

Shannon, C.E. (1950). Programming a Computer for Playing Chess. *Philosophical Magazine*, Vol. 41, No. 4, pp. 256-275.

Shortliffe, E.H. (1976). *MYCIN: Computer-Based Medical Consultations*, Elsevier Press, New York, NY.

Turban, E. (1995). *Decision Support and Expert Systems: Management Support Systems*, Prentice-Hall International, Englewood Cliffs, NJ.

Turing, A.M. (1950). Computing Machinery and Intelligence. *Mind*, Vol. 59, pp. 433-460.

van Melle, W. (1979). A Domain Independent Production-Rule System for Consultation Programs. *Proceedings of the IJCAI 6*, pp. 923-925.

van Melle, W., Shortliffe, E.H. and Buchanan B.G. (1981). EMYCIN: A Domain-Independent System that Aids in Constructing Knowledge-Based Consultation Programs. *Machine Intelligence, Infotech State of the Art Report 9*, no. 3.

Waterman, D.A. (1986). *A Guide to Expert Systems*, Addison-Wesley Publishing Company, Reading, Massachusetts.

Yager, R.R., and Zadeh, L.A. (1994). *Fuzzy Sets, Neural Networks and Soft Computing*, Van Nostrand Reinhold, New York.

Zadeh, L. (1965). Fuzzy Sets, *Information and control*, Vol.8, No.3, pp. 338 - 353.

Zahedi, F. (1993). *Intelligent Systems for Business: Expert Systems with Neural Networks*. Belmont, CA: Wadsworth.