**LECTURE 6**

**ARTIFICIAL INTELLIGENCE- 1**

Having examined the history of computer applications we are naturally attracted to an examination of their future limits, not in computing capacity but rather what kinds of things computers can and perhaps cannot do. Before we get too far I need to remind you that computers manipulate symbols, not information; we are simply unable to say, let alone write a program for, what we mean by the word "information". We each believe we know what the word means, but hard thought on your part will convince you that it is a fuzzy concept at best, that you cannot give a definition that can be converted into a program.

Although Babbage and Lady (Ada) Lovelace both considered slightly some of the limitations of computers, the exploration of the limits of computers really began in the late 1940's and early 50's by, among others, Newell and Simon at RAND. For example they looked at puzzle solving, such as the classic cannibals and missionaries problem. Could machines solve them? And how would they do it? They examined the protocols people used as they solved such problems, and tried to write a program that would produce similar results. You should not expect exactly the same result as generally no two people reported exactly the same steps in the same order of their thought processes, rather the program was to produce a similar looking pattern of reasoning. Thus they tried to model the way people did such puzzles and examine how well the model produced results resembling human results, rather than just solve the problem.

They also started the General Problem Solver (GPS) with the idea that given about 5 general rules for solving problems they could then give the details of the particular area of a problem and the computer program would solve the problem. It didn't work too well, though very valuable by-products did come from their work such as list processing. To continue with this problem solving approach that they started, after their initial attack on general problem solving (which certainly promised to alleviate the programming problem to a fair extent) it was dropped, more or less, for a decade, and when revived the proposal was that about 50 general rules would be needed. When that did not work, another decade and the proposal with 500 general rules, and another decade, now under the title of rule based logic and they are sometimes at 5000 rules, and I have even heard of 50,000 as the number of rules for some areas.

There is now a whole area known as Expert Systems. The idea is that you talk with some experts in a field, extract their rules, put these rules into a program, and then you have an expert! Among other troubles with this idea is that in many fields, especially in medicine, the world famous experts are in fact not that much better than the beginners! It has been measured in many different studies! Another trouble is that experts seem to use their subconscious, and they can only report their conscious experience in making a diagnosis. It has been estimated that it takes about 10 years of intensive work in a field to become an expert, and in this time many, many patterns are apparently laid down in the mind from which the expert then makes a subconscious initial choice of how to approach the problem as well as the subsequent steps to be used.

In some areas rule based logic has had spectacular successes, and in some apparently similar areas there were plain failures, which indicates that success depends on a large element of luck; they still do not have a firm basic understanding of when the method of rule based logic will or will not work, nor how well it will work.

In Lecture 1 I already brought up the topic that perhaps everything we "know" <u>cannot</u> be put into words (instructions) - cannot in the sense of impossible and not in the sense we are stupid or ignorant. Some of the features of Expert Systems we have found certainly strengthen this opinion.

After quite a few years the field of the limits of intellectual performance by machines acquired the dubious title of <u>artificial intelligence</u> (AI), which does <u>not</u> have a single meaning. First, it is a variant on the question,

**CAN MACHINES THINK?**

While this is a more restricted definition than is artificial intelligence, it has a sharper focus and is a good substitute in the popular mind. This question is important to you because if you believe that computers cannot think then as a prospective leader you will be slow to use computers to advance the field by your efforts, but if you believe that of course computers can think then you are very apt to fall into a first class failure! Thus you cannot afford to either believe or disbelieve - you must come to your own terms with the vexing problem, "To what extent can machines think?"

Note, first, that it really is misstated - the question seems to be more, "Can we write programs that will produced 'thinking' from a von Neumann type machine?" The reason for the hedge is that there are arguments that modern neural nets, when not simulated on a digital computer, might be able to do what no digital computer can do. But then again they might not. It is a problem that we will look into at a later stage when we have more technical facts available.

While the problem of AI can be viewed as, "Which of all the things humans do can machines also do?" I would prefer to ask the question in another form, "Of all of life's burdens, which are those that machines can relieve, or significantly ease, for us?" Note that while you tend to automatically think of the material side of life, pacemakers are machines connected directly to the human nervous system and help keep many people alive. People who say that they do not want their life to depend on a machine seem quite conveniently to forget this. It seems to me that in the long run it is on the intellectual side of life that machines can most contribute to the quality of life.

Why is the topic of artificial intelligence important? Let me take a specific example of the need for AI. Without defining things more sharply, (and without defining either thinking or what a machine is there can be no real proof one way or the other), I believe that very likely in the future we will have vehicles exploring the surface of Mars. The distance between Earth and Mars at times may be so large that the signaling time round trip could be 20 or more minutes. In the exploration process the vehicle must, therefore, have a fair degree of local control. When having passed between two rocks, turned a bit, and then found the ground under the front wheels was falling away, you will want prompt, "sensible" action on the part of the vehicle. Simple, obvious things like backing up will be inadequate to save it from destruction, and there is not time to get advice from Earth; hence some degree of "intelligence" should be programmed into the machine.

This is not an isolated situation; it is increasingly typical as we use computer driven machines to do more and more things at higher and higher speeds. You cannot have a human backup - often because of the boredom factor that humans suffer from. They say that piloting a plane is hours of boredom and seconds of sheer panic - not something that humans were designed to cope with, though they manage to a reasonable degree. Speed of response is often essential. To repeat an example, our current fastest planes are basically unstable and have computers to stabilize them, millisecond by millisecond, which no human pilot could handle; the human can only supply the strategy in the large and leave the details in the small to the machine.

I earlier remarked on the need to get at least some understanding of what we mean by "a machine" and by "thinking". We were discussing these thing at Bell Telephone Laboratories in the late 40's and some one said that a machine could not have organic parts, upon which I said that definition excluded any wooden parts! The first definition was retracted, but to be nasty I suggested that in time we might learn how to remove a large part of a frog's nervous system and keep it alive. If we found how to use it for a storage mechanism, would it be a machine or not? If we used it as an a content addressable storage how would you feel about it being a "machine"?

In the same discussion, on the thinking side, a Jesuit trained engineer gave the definition, "Thinking is what humans can do and machines cannot do." Well, that solves the problem once and for all, apparently. But do you like the definition? Is it really fair? As we pointed out to him then, if we start with some obvious difference at present then with improved machines and better programming we may be able to reduce the difference, and it is not clear that in the long run there would be any difference left.

Clearly we need to define "thinking". Most people want the definition of thinking to be such that they can think but that stones, trees, and such things, cannot think. But people vary to the extent that they will or will not include the higher levels of animals. People often make the mistake of saying, "Thinking is what Newton and Einstein did." but by that definition most of us cannot think - and usually we don't like that conclusion! Turing, in coping with the question in a sense evaded it and made the claim that if at the end of one teletype line there was a human and at the end of another teletype line there was a suitably programmed machine, and if the average human could not tell the difference then that was a proof of "thinking" on the part of the machine (program).

The Turing test is a popular approach, but it flies in the face of the standard scientific method which starts with the easier problems before facing the harder ones. Thus I soon raised the question with myself, "What is the smallest or close to the smallest program that I would believe could think?" Clearly if the program were divided into two parts then neither piece could think. I tried thinking about it each night as I put my head on the pillow to sleep, and after a year of considering the problem and getting nowhere I decided that it was the wrong question! Perhaps "thinking" is not a yes-no thing, but maybe it is a matter of degree.

Let me digress and discuss some of the history of chemistry. It was long believed that organic compounds could only be made by living things, that there was a <u>vitalistic</u> aspect in living things but not in inanimate things such as stones and rocks. But around 1823 a chemist named Wohler synthesized urea, a standard by-product of humans. This was the beginning of making organic compounds in test tubes. Still, apparently even as late as 1850, the majority of chemists were holding to the <u>vitalistic</u> theory that only living things could make organic compounds. Well, you know that from this attitude we have gone to the other extreme and now most chemists believe that in principle <u>any</u> compound that the body can make can also be made in the lab - but of course there is no proof of this, nor could there ever be. The situation is that they have an increasing ability to make organic compounds, and see no reason that they cannot make any compound that can exist in Nature as well as many that do not. Chemists have passed from the <u>vitalistic</u> theory of chemistry to the opposite extreme of a <u>non-vitalistic</u> theory of chemistry.

Religion unfortunately enters into discussions of the problem of machine thinking, and hence we have both vitalistic and non-vitalistic theories of "machines vs. humans". For the Christian religions their Bible says that, "God made Man in His image." If we can in turn create machines in our image then we are in some sense the equal of God, and this is a bit embarrassing! Most religions, one way or the other, make man into more than a collection of molecules, indeed man is often distinguished from the rest of the animal world by such things as a soul, or some other property. As to the soul, in the Late Middle Ages some people, wanting to know when the soul departed from the dead body, put a dying man on a scale and watched for the sudden change in weight - but all they saw was a slow loss as the body decayed - apparently the soul, which they were sure the man had, did not have material weight.

Even if you believe in evolution, still there can be a moment when God, or the gods, stepped in and gave man special properties that distinguish him from the rest of living things. This belief in an essential difference between man and the rest of the world is what makes many people believe that machines can never, unless we ourselves become like the gods, be the same as a human in such details as thinking, for example. Such people are forced, like the above mentioned Jesuit trained engineer, to make the definition of thinking to be what machines cannot do. Usually it is not so homestly stated as he did, rather it is disguised somehow behind a facade of words, but the intention is the same!

Physics regards you as a collection of molecules in a radiant energy field and there is, in strict physics, <u>nothing else</u>. Democritus (b. around 460) said in ancient Greek times, "All is atoms and void". This is the stance of the <u>hard AI</u> people; there is no essential difference between machines and humans, hence by suitably programming machines then machines can do anything that humans can do. Their failures to produce thinking in significant detail is, they believe, merely the failure of programmers to understand what they are doing, and not an essential limitation.

At the other extreme of the AI scale, some of us, when considering our own feelings, believe that we have <u>self-awareness</u> and <u>self-consciousness</u> - though we are not able to give satisfactory tests to prove that these things exist. I can get a machine to print out, "I have a soul.", or "I am self-aware.", or "I have self-consciousness.", and you would not be impressed with such statements from a machine. But from humans you are inclined to give greater credence to such remarks, based on the belief that you, by introspection, feel that you have such properties (things), and you have learned by long experience in life that other humans are similar to you - though clearly racism still exists which asserts that there are differences - me being always the better person!

We are at a stalemate at this point in the discussion of AI; we can each assert as much as we please, but it proves nothing at all to most people. So let us turn to the record of AI successes and failures.

AI people have always made extravagant claims that have not been borne out - not even closely in most cases. Newell and Simon in 1958 predicted that in 10 years the next world champion in chess would be a computer program. Unfortunately similar, as yet unrealized, claims have been made by most of the AI leaders in the public eye. Still, startling results have been produced.

I must again digress, this time to point out why game playing has such a prominent role in AI research. The rules of a game are clear beyond argument, and success or failure are also - in short the problem is well defined in any reasonable sense. It is not that we particularly want machines to play games, but that they provide a very good testing ground of our ideas on how to get started in AI.

Chess, from the beginning, was regarded as a very good test since it was widely believed at that time that chess requires thinking <u>beyond any doubt</u>. Shannon proposed a way of writing chess playing programs (we call them chess playing machines but it is really mainly a matter of programming). Los Alamos, with a primitive MANIAC machine tried 6x6 chess boards, dropping the two bishops on each side, and got moderate results. We will return to the history of chess playing programs later.

Let us examine how one might write a program for the much simpler game of three dimensional tic-tac-toe. We set aside simple two dimensional tic-tac-toe since it has a known strategy for getting a draw, and there is no possibility of win against a prudent player. Games that have a known strategy of playing simply are not exhibiting thinking - so we believe at the moment.

As you examine the 4x4x4 cube there are 64 squares, and 76 straight lines through them. Any one line is a <u>win</u> if you can get all four of the positions filled with your pieces. You next note that the 8 corner locations, and the 8 center locations, all have more lines through them than the others; indeed there is an inversion of the cube such that the center points go to the corners and the corners go to the center while preserving all straight lines - hence a duality that can be exploited if you wish.

For a program to play 4x4x4 tic-tac-toe it is first necessary to pick legal moves. Then in the opening moves you tend to place your pieces on these "hot" spots, and you use a random strategy since otherwise, since if you play a standard game then the opponent can slowly explore it until a weakness is uncovered that can be systematically exploited. This use of randomness, when there are essentially indifferent moves, is a central part of all game playing programs.

We next formulate some rules to be applied sequentially.

1. if you have 3 men on a line and it is still "open" then play it and win.

2. if you have no immediate win, and if the opponent has 3 men on a line, then you must block it.

3. if you have a fork, (Figure 6-1), take it since then on the next move you have a win, as the opponent cannot win in one move.

4. If the opponent has a fork you must block it.

After this there are apparently no definite rules to follow in making your next move. Hence you begin to look for "forcing moves", ones that will get you to some place where you have a winning combination. Thus 2 pieces on an "open" line means that you can place a third and the opponent will be forced to block the line, (but you must be careful that the blocking move does not produce three in a line for the opponent and force you to go on the defensive). In the process of making several forcing moves you may be able to create a fork, and then you have win! But these rules are vague. Forcing moves that are on "hot" places and where the opponent's defense must be on a "cool" places seem to favor you, but does not guarantee a win. In starting a sequence of forcing moves, if you lose the initiative, then almost certainly the opponent can start a sequence of forcing moves on you and gain a win. Thus when to go on the attack is a touchy matter; too soon and you lose the initiative, too late and the opponent starts and wins. It is not possible, so far as I know to give an exact rule of when to do so.

This is the standard structure of a program to play a game on a computer. Programs must first require that you check that the move is legal before any other step, but that is a minor detail. Then there is usually a set of more or less formal rules to be obeyed, followed by some much vaguer rules, Thus a game program has a lot of <u>heuristics</u> in it, (heuristic - to invent or discover), moves that are plausible and likely to lead you to a win, but are <u>not</u> guaranteed to do so.

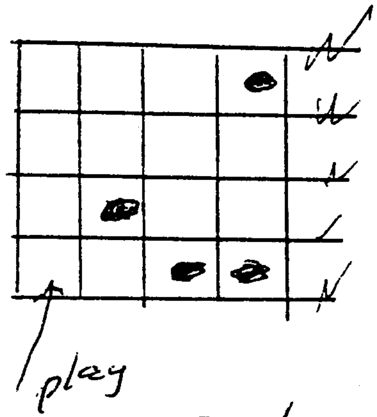
Early in the field of AI Art Samuel, then at IBM, wrote a checker playing program, checkers being thought to be easier than chess which had proved to be a real stumbling block. The formula he wrote for playing checkers had a large number of rather arbitrary parameters in the weighting functions for making decisions, such as for control of the center, passed pieces, kings, mobility, pinned pieces, etc. Samuel made a copy of the program and then slightly altered one (or more) of these parameters. Then he made one formula play, say, ten games against the other, and the formula that won the most games was clearly (actually only probably) the better program. He went on perturbing the same parameters until he came to a local optimum, whereupon he shifted to other parameters. Thus he went around and around, repeatedly using the same parameters, gradually emerging with a significantly better checker playing program - certainly much better than was Samuel himself. The program even beat a Connecticut State checker champion!

Is it not fair to say, "The program learned from experience"? Your immediate objection is that there was a program telling the machine how to learn. But when you take a course in Euclidean geometry is not the teacher putting a similar learning program into you? Poorly, to be sure, but is that not, in a real sense, what a course in geometry is all about? You enter the course and cannot do problems; the teacher puts into you a program and at the end of the course you can solve such problems. Think it over carefully. If you deny that the machine learns from experience because you claim that the program was told (by the human programmer) how to do improve its performance, then is not the situation much the same with you, except that you are born with a somewhat larger initial program compared to the machine when it leaves the manufacturer's hands? Are you sure you are not merely "programmed" in life by what by chance events happen to you?

We are beginning to find that not only is intelligence not adequately defined so that arguments can be settled scientifically, but a lot of other associated words like, computer, learning, information, ideas, decisions, (hardly a mere branching of a program, though branch points are often called decision points to make the programmers feel more important), expert behavior - all are a bit fuzzy in our minds when we get down to the level of testing them via a program in a computer. Science has traditionally appealed to experimental evidence and not idle words, and so far science seem to have been more effective than philosophy in improving our way of life. The future can, of course, be different.

In this lecture we have "set the stage" for a further discussion of AI. We have also claimed that it is not a topic you can afford to ignore. Although there seems to be no hard, factual results, and perhaps there can never be since the very words are ill-defined and are open to modification and various interpretations, still you must come to grips with it. In particular, when a program is written that does meet some earlier specification for a reasonable test of computer learning, originality, creativity, or intelligence, then it is promptly seen by many people that the test had a mechanical solution. This is true even if random numbers are involved, and given the same test twice the machine will get a solution that differs slightly from the earlier one, much as humans seldom play exactly the same game of chess twice in a row. What is a reasonable, practical test of machine learning? Or are you going to claim, as the earlier cited Jesuit trained engineer did, that by definition learning, creativity, originality, and intelligence are what machines cannot do? Or are you going to try to hide this blatant statement and conceal it in some devious fashion that does not really alter the situation?

In a sense you will never really grasp the whole problem of AI until you get inside and try your hand at finding what you mean and what machines can do. Before the checker playing program that learned was exposed in simple detail, you probably thought that machines could not learn from experience - now you may feel that what was done was not learning but clever cheating, though clearly the program modified its behavior depending on its experiences. <u>You must struggle with your own beliefs if you are to make any progress in understanding the possibilities and limitations of computers in the intellectual area</u>. To do this adequately you must formalize your beliefs and then criticize them severely, arguing one side against the other, until you have a fair idea of the strengths and weakness of both sides. Most students start out anti-AI; some are pro AI; and if you are either one of these then you must try to undo your biases in this important matter. In the next Lecture we will supply more surprising data on what machines have done, but you must make up your own mind on this important topic. False beliefs will mean that you will not participate significantly in the inevitable and extensive computerization of your organization and society generally. In many senses the computer revolution has only begun!




A Fork

Figure 6-1