Artificial Intelligence and Neural Networks The Legacy of Alan Turing and John von Neumann

Heinz Muehlenbein

Fraunhofer Institut Autonomous intelligent Systems
Schloss Birlinghoven 53757 Sankt Augustin, Germany
heinz.muehlenbein@online.de, http://www.ais.fraunhofer.de/~muehlen

Abstract— The work of Alan Turing and John von Neumann on machine intelligence and artificial automata is reviewed. Turing's proposal to create a child machine with the ability to learn is discussed. Von Neumann's had doubts that with teacher based learning it will be possible to create artificial intelligence. He concentrated his research on the issue of complication, probabilistic logic, and self-reproducing automata. The problem of creating artificial intelligence is far from being solved. In the last sections of the paper I review the state of the art in probabilistic logic, complexity research, and transfer learning. These topics have been identified as essential components of artificial intelligence by Turing and von Neumann.

I. INTRODUCTION

Computer based research on machine intelligence started about 60 years ago, parallel to the construction of the first electronic computers. Therefore it seems to be time again to compare todays state-of-the art with thoughts and proposals at the very beginning of the computer age. I have chosen Alan Turing and John von Neumann as the most important representatives of the first concepts of machine intelligence. Both researchers actually designed electronic computers, but they also reflected about what the new electronic computers could be expected to solve in addition to numerical computation. Both discussed intensively the problem how the performance of the machines will ultimately compare to the power of the human brain.

In this paper I will first review the work of Alan Turing, contained in his seminal paper "Computing Machinery and Intelligence" (17) and in the not so well known paper "Intelligent Machinery" (18). Then I will discuss the most important paper of John von Neumann concerning our subject "The General and Logical Theory of Automata" (22). All three papers have been written before the first electronic computers became available. Turing even wrote programs for paper machines.

I will describe the thoughts and opinions of Turing and von Neumann in detail, without commenting them using todays knowledge. Then I will try to evaluate their proposals in answering the following questions

- What are their major ideas for creating machine intelligence?
- Did their proposals lack important components we see as necessary today?
- What are the major problems of their designs and do their exists solutions today?

This paper extends my research started in (12).

II. TURING AND MACHINE INTELLIGENCE

The first sentences of the paper "Computing machinery and intelligence" have become famous. "I propose to consider the question "Can machines think?" This should begin with definitions of the meaning of the terms "machine" and "think"....But this is absurd. Instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words. The new form of the question can be described in terms of a game which we call the imitation game."

The original definition of the imitation game is more complicated than what is today described as the Turing test. Therefore I describe it shortly. It is played with three actors, a man (A), a woman (B) and an interrogator (C). The object of the game for the interrogator is to determine which of the other two is the man and which is the woman. It is A's objective in the game to try and cause C to make the wrong identification. Turing then continues: "We now ask the question "What will happen when a machine takes the part of A in the game?" Will the interrogator decide wrongly as often when the game is played as this as he does when the game is played between a man and a woman? These questions will replace our original "Can machines think".

Why did Turing not define just a game between a human and a machine trying to imitate a human, as the Turing test is described today? Is there an

additional trick in introducing gender into the game? There has been a quite a lot of discussions if this game characterizes human intelligence at all. Its purely behavioristic definition leaves out any attempt to identify important components which together produce human intelligence. I will not enter this discussion here, but just state the opinion of Turing about the outcome of the imitation game.

"It will simplify matters for the readers if I explain first my own beliefs in the matter. Consider first the more accurate form of the question. I believe that in about fifty years' time it will be possible to programme computers with a storage capacity of about 10^9 bits to make them play the imitation game so well that an average interrogator will not have more than 70% chance of making the right identification after five minutes of questioning."

The accurate form of the question is obviously artificial definite: Why a 70% chance, how often has the game to be played, why a duration of five minutes? In the next section I will discuss what Turing lead to predict 50 years. The prediction is derived in section 7 of his paper (17).

III. Turing's construction of an intelligent machine

In section 7 Turing discusses how to build an intelligent machine. In the sections before Turing mainly refuses general philosophical arguments against the possibility of constructing intelligent machines. "The reader will have anticipated that I have no very convincing argument of a positive nature to support my views. If I had I should not have taken such pains to point out the fallacies in contrary views. Such evidence as I have I shall now give." What is Turing's evidence?

"As I have explained, the problem is mainly one of programming. Advances in engineering will have to be made too, but it seems unlikely that these will not be adequate for the requirements. Estimates of the storage capacity of the brain vary from 10^{10} to 10^{15} binary digits.[1] I incline to the lower values and believe that only a small fraction is used for the higher types of thinking. Most of it is probably used for the retention of visual impressions. I should be surprised if more than 10^9 was required for satisfactory playing of the imitation game. Our problem then is to find out how to programme these machines to play the game. At my present rate of working I produce about a thousand digits of programme a day, so that about sixty workers, working steadily through fifty years might accomplish the job, if

nothing went into the wastepaper basket."

The time to construct a machine which passes the imitation game is derived from an estimate of the storage capacity of the brain [2] and the speed of programming. Turing did not see any problems in creating machine intelligence purely by programming, he just found it too time consuming. So he investigated if there exist more expeditious methods. He observed:
"In the process of trying to imitate an adult human mind we are bound to think a good deal about the process which has brought it to the state that it is in. We may notice three components.

1) The initial state of the brain, say at birth.
2) The education to which it has been subjected.
3) Other experience, not to be described as education, to which it has been been subjected.

Instead of trying to produce a programme to simulate an adult mind, why not rather try to produce one which simulates the child's...Presumably the child brain is something like a notebook. Rather little mechanism, and lots of blank sheets. Our hope is that there is so little mechanism in the child brain that something like it can easily be programmed. The amount of work in the education we can assume, as a first approximation, to be much the same as for the human child."

A. Turing on learning and evolution

In order to achieve a greater efficiency in constructing a machine with human like intelligence, Turing divided the problem into two parts

- The construction of a child brain
- The development of effective learning methods

Turing notes that the two parts remain very closely related. He proposes to use experiments: teaching a child machine and see how well it learns. One should then try another and see if it is better or worse. "There is an obvious connection between this process and evolution, by the identifications

- structure of the machine = hereditary material
- changes of the machine = mutations
- Natural selection = judgment of the experimenter

Survival of the fittest is a slow process of measuring advantages. The experimenter, by the exercise of intelligence, should be able to speed it up."

Turing then discusses learning methods. He notes ((17),p.454):"We normally associate the use of punishments and rewards with the teaching process...The machine has to be so constructed that events which shortly proceeded the occurrence of

---

[1]At this time the number of neurons was estimated as being between 10^{10} to 10^{15}. This agrees with the estimates using todays knowledge.

[2]It was of course a big mistake to set the storage capacity equal to the number of neurons! We will later show that von Neumann estimated the storage capacity of the brain to be about 10^{20}.

a punishment signal are unlikely to be repeated, whereas a reward signal increased the probability of repetition of the events which lead to it." But Turing observes the major drawback of this method:"The use of punishments and rewards can at best be part of the teaching process. Roughly speaking, if the teacher has no other means of communicating to the people, the amount of information which can reach him does not exceed the total number of rewards and punishments applied."

In order to speed up learning Turing demanded that the child machine should understand some language. In the final pages of the paper Turing discusses the problem of the complexity the child machine should have. He proposes to try two alternatives: either to make it as simple as possible to allow learning or to include a complete system of logical inference. He ends his paper with the remarks: "Again I do not know the answer, but I think both approaches should be tried. We can see only see a short distance ahead, but we can see plenty there that needs to be done."

B. Turing and neural networks

In the posthumously published paper *Intelligent Machinery* (18) Turing describes additional details how to create an intelligent machine. First he discusses possible components of a child machine. He introduces *unorganized machines* of type A,B, and P. A and B are artificial neural networks with random connections. They are made up from a rather large number N of similar units, which can be seen as binary neurons. Each unit has two input terminals and one output terminal which can be connected to the input terminals of 0 (or more) other units. The connections are chosen at random. All units are connected to a central synchronizing unit from which synchronizing pulses are emitted. Each unit has two states. The dynamics is defined by the following rule:

The states from the units from which the input comes are taken from the previous moment, multiplied together and the result is subtracted from 1.

This rule gives an unusual transition table. I doubt that this rule is powerful enough. The state of the network is defined by the states of the units. Note that the network might have lots of loops, it continually goes through a number of states until a period begins. The period cannot exceed 2^N cycles. In order to allow learning the machine is connected with some input device which can alter its behavior. This might be a dramatic change of the structure, or changing the state of the network. Maybe Turing had the intuitive feeling that the basic transition of the type A machine is not enough, therefore

he introduced the more complex B-type machine. I will not describe this machine here, because neither for the A or the B machine Turing defined precisely how learning can be done.

A learning mechanism is introduced with the third machine, called a P-type machine. The machine is an automaton with a number of N configurations. There exist a table where for each configuration is specified which action the machine has to take. The action may be either

1) To do some externally visible act $A_1, \ldots A_k$
2) To set a memory unit M_i

The reader should have noticed that the next configuration is not yet specified. Turing surprisingly defines: The next configuration is always the remainder of $2s$ or $2s + 1$ on division by N. These are called the alternatives 0 and 1. The reason for this definition is the learning mechanism Turing defines. At the start the description of the machine is largely incomplete. The entries for each configuration might be in five states, either U (uncertain), or T0 (try alternative 0), T1 (try alternative 1), D0 (definite 0) or D1 (definite 1).

Learning changes the entries as follows: If the entry is U, the alternative is chosen at random, and the entry is changed to either T0 or T1 according to whether 0 or 1 was chosen. For the other four states, the corresponding alternatives are chosen. When a pleasure stimulus occurs, state T is changed to state D, when a pain stimulus occurs, T is changed to U. Note that state D cannot be changed. The proposed learning method sounds very simple, but Turing surprisingly remarked:

I have succeeded in organizing such a (paper) machine into a universal machine.

Today this universal machine is called the Turing Machine. Turing even gave some details of this particular P-type machine. Each instruction consisted of 128 digits, forming four sets of 32 digits, each of which describes one place in the main memory. These places may be called P,Q,R,S. The meaning of the instruction is that if p is the digit at P and q that at Q then $1 - pq$ is to be transferred to position R and the next instruction will be found at S. The universal machine is not the solution to the problem, it has to be programmed!

C. Discipline and initiative

We now turn to the next important observation of Turing. Turing notes that punishment and reward are very slow learning techniques. So he requires:

If the untrained infant's mind is to become an intelligent one, it must acquire both discipline and initiative.

Discipline means strictly obeying the punishment and reward. But what is initiative? The definition of initiative is typical of Turing's behavioristic attitude. "Discipline is certainly not enough in itself to produce intelligence. That which is required in addition we call initiative. This statement will have to serve as a definition. Our task is to discover the nature of this residue as it occurs in man, and to try and copy it in machines."

With only a paper computer available Turing was not able to investigate the subject initiative further. Nevertheless he made the bold statement (18): "A great positive reason for believing in the possibility of making thinking machinery is the fact that it is possible to make machinery to imitate any small part of a man. One way of setting about our task of building a thinking machine would be to take a man as a whole and to try to replace all parts of him by machinery...Thus although this method is probably the 'sure' way of producing a thinking machine it seems to be altogether too slow and impracticable. Instead we propose to try and see what can be done with a 'brain' which is more or less without a body providing, at most organs of sight, speech, and hearing. We are then faced with the problem of finding suitable branches of thought for the machine to exercise its powers in."

Turing mentions the following fields as promising:
- Various games, e.g. chess, bridge
- The learning of languages
- Translation of languages
- Cryptography
- Mathematics

Turing remarks: "The learning of languages would be the most impressive, since it is the most human of these activities. This field seems however to depend rather too much on sense organs and locomotion to be feasible." Turing seems here to have forgotten that language learning is necessary for his imitation game!

IV. VON NEUMANN'S LOGICAL THEORY OF AUTOMATA

Alan Turing was for a short time in 1938 assistant of John von Neumann. But later they worked completely independent from each other, not knowing the thoughts the other had concerning the power of the new electronic computers. A condensed summary of the research of John von Neumann concerning machine intelligence, or in his more low-key term "artificial automata", is contained in his paper "The General and Logical Theory of Automata" (22). This paper was presented in 1948 at the Hixon symposium on: *Cerebral mechanism of behavior*. Von Neumann was the only computer scientist at this symposium. His invitation indicates his interdisciplinary research. This is clearly expressed in the first page:

Natural organisms are, as a rule, much more complicated and subtle, and therefore much less well understood in detail, than are artificial automata. Nevertheless, some of the regularities which we observe in the former may be quite instructive in our thinking and planning of the latter; and conversely, a good deal of our experiences and difficulties with our artificial automata can be to some extend projected on our interpretations of natural organisms.

Von Neumann notices three major limits of the present size of artificial automata
- The size of componentry
- The limited reliability
- The lack of a logical theory of automata

There have been tremendous achievements in the first two areas. Therefore I will concentrate on the theory problem. The new theory of logical automata has to investigate the following topics.

"The logic of automata will differ from the present system of formal logic in two relevant respects.
1) *The actual length of "chains of reasoning", that is, of the chains of operations, will have to be considered.*
2) *The operations of logic will all have to be treated by procedures which allow exceptions with low but non-zero probabilities.*

Von Neumann tried later to formulate probabilistic logic. His results appeared in (23). But this research was more or less a dead end, because von Neumann did not abstract enough from the logical hardware components and introduced time into the analysis. But in (22) he remarked prophetically:
This new system of formal logic will move closer to another discipline which has been little linked in the past with logic. This is thermodynamics, primarily in the form it was received from Boltzmann, and is that part of theoretical physics which comes nearest in some of its aspects to manipulating and measuring information.

A. McCulloch-Pitts theory of formal neural networks

In (9) McCulloch and Pitts had described the brain by a formal neural network, consisting of interconnected binary neurons. Von Neumann summarizes their major result follows: "The "functioning" of such a network may be defined by singling out some of the inputs of the entire system and some of its outputs, and then describing what original stimuli on the former are to cause what ultimate stimuli of the latter. McCulloch and Pitts' important result is that any functioning in this sense which can be defined at all logical, strictly, and unambiguously in a finite number of words can also be realized by such a formal system."

McCulloch and Pitts had derived this result by showing that their formal neural network connected to an infinite

tape is equivalent to a Turing machine. But even given this result, von Neumann observes that at least two problems remain

1) Can the network be realized within a practical size?
2) Can every existing mode of behavior really be put completely and unambiguously into word?

Von Neumann informally discusses the second problem, using the example visual analogy. He remarks prophetically:

There is no doubt that any special phase of any conceivable form of behavior can be described "completely and unambiguously" in words.... It is, however an important limitation, that this applies only to every element separately, and it is far from clear how it will apply to the entire syndrome of behavior.

This severe problem has not been noticed by Turing. Using the example visual analogy von Neumann argues: "One can start describing to identify any two rectilinear triangles. These could be extended to triangles which are curved, whose sides are only partially drawn etc... We may have a vague and uncomfortable feeling that a complete catalogue along such lines would not only be exceedingly long, but also unavoidingly indefinite at its boundaries. All of this, however, constitutes only a small fragment of the more general concept of identification of analogous geometrical objects. This, in turn, is only a microscopic piece of the general concept of visual analogy." Thus von Neumann comes to the conclusion:

Now it is perfectly possible that the simplest and only practical way to say what constitutes a visual analogy consists in giving a description of the connections of the visual brain....It is not at all certain that in this domain a real object might not constitute the simplest description of itself.

Von Neumann ended this section with the sentence: "The foregoing analysis shows that one of the relevant things we can do at this moment is to point out the directions in which the real problem does not lie." Instead of investigating the above complexity issue directly, von Neumann turned to the more fundamental problem of the complexity needed for automata solving difficult problems.

B. Complication and self-reproduction

Von Neumann starts the discussion of complexity with the observation that if an automaton has the ability to construct another one, there must be a decrease in complication. In contrast, natural organisms reproduce themselves, that is, they produce new organisms with no

decrease in complexity. So von Neumann tries to construct a general artificial automata which could reproduce itself. The famous construction works as follows:

1) A general constructive machine, A, which can read a description $\Phi(X)$ of another machine, X, and build a copy of X from this description:

$$A + \Phi(X) \rightsquigarrow X$$

2) A general copying machine, B. which can copy the instruction tape:

$$B + \Phi(X) \rightsquigarrow \Phi(X)$$

3) A control machine, C, which when combined with A and B, will first activate B, then A, link X to $Phi(X)$ and cut them loose from A+B+C

$$A + B + C + \Phi(X) \rightsquigarrow X + \Phi(X)$$

Now choose X to be A+B+C

$$A+B+C+\Phi(A+B+C) \rightsquigarrow A+B+C+\Phi(A+B+C)$$

4) It is possible to add the description of any automaton D

$$A + B + C + \Phi(A + B + C + D) \rightsquigarrow A + B + C + D \\ +\Phi(A + B + C + D)$$

Now allow mutation on the description $\Phi(A+B+C+D)$

$$A + B + C + \Phi(A + B + C + D') \rightsquigarrow A + B + C + D' \\ +\Phi(A + B + C + D')$$

Mutation at the D description will lead to a different self-reproducing automaton. This might allow to simulate some kind of evolution as seen in natural organisms.

Von Neumann later constructed a self-reproducing automata which consisted of 29 states (24). This convinced von Neumann that complication can also be found in artificial automata. Von Neumann ends the paper with the remark:

This fact, that complication, as well as organization, below a critical level is degenerative, and beyond that level can become self-supporting and even increasing, will clearly play an important role in any future theory of the subject.

V. Discussion of the designs of Turing and von Neumann

I have reviewed only a small part of the research of Turing and von Neumann concerning machine intelligence and artificial automata. But one observation strikes immediately: both researchers investigated the problem of machine intelligence on a very broad scale. The main emphasis of Turing was the design of efficient learning schemes. For Turing it was obvious that only

by learning and creating something like a child machine an intelligent machine could be developed. The attitude of Turing was purely that of a computer scientist. Using mainly an estimate of the memory capacity of the human brain, he firmly believed that machine intelligence equal to or surpassing human intelligence can be created.

Von Neumann's approach was more interdisciplinary, using also results from the analysis of the brain. He had a similar goal, but he was much more cautious concerning the possibility to create an automaton with intelligence. He investigated important problems one by one which appeared him on the road to machine intelligence.

Both researchers investigated formal neural networks as a basic component of an artificial brain. This component was not necessary for the design, it was used only to show that the artificial automata could have a similar organization as the human brain. Both researchers ruled out that a universal theory of intelligence could be found, which would make it possible to program a computer according to this theory. So Turing proposed to use *learning* as the basic mechanism, von Neumann *self-reproducing automata*. Von Neumann was more radical because he was convinced that learning leads to the *curse of infinite enumeration*. Turing also saw the limitations of teacher based learning by reward and punishment, therefore he required that the machine needs *initiative* in addition.

The designs of Turing and von Neumann contain all components considered necessary today for machine intelligence. Turing ended his investigation with the problem of initiative, which is still an unresolved issue today. Von Neumann's idea to use self-reproducing automata has not yet lead to an automata with interesting behavior. The problem of von Neumann's approach is the following: In order that his automaton does something besides reproducing one has to input a program D for each task. How can the machine develop more complex programs starting with an initial program?
There seem to be no major failure in their designs, but at least two major issues are not yet resolved
- The memory capacity of the brain
- Can every problem which is computable be learned from examples?

I will discuss the capacity problem first.

VI. MEMORY CAPACITY OF THE BRAIN

Von Neumann also estimated the capacity of the brain. His estimate can be found in the book "The computer and the brain" ((23),p. 63)

"However, certain rough orienting estimates can, nevertheless, be arrived at. Thus the standard receptor (neuron) would seem to accept 14 distinct digital impressions per second. Allowing 10^{10} nerve cells gives a total input of $14 * 10^{10}$ bits per second. Assuming further, for which there is some evidence, that there is no true forgetting in the nervous system - an estimate for the entirety of a normal human lifetime can be made. Putting the latter equal to, say, 60 years $\approx 2 * 10^9$ seconds, the total required memory capacity would turn out to be $2.8 * 10^{20}$."

Note that this estimate is 10^{10} times larger than the estimate of Turing! There is still no agreement on the memory capacity of the brain. The brain is highly redundant and not well understood: the mere fact that a great mass of synapses exists does not imply that they are in fact all contributing to memory capacity.

A totally different method to estimate the capacity has been pursued by Landauer (4). He reviewed and quantitatively analyzed experiments by himself and others in which people were asked to read text, look at pictures, and hear words, short passages of music, sentences, and nonsense syllables. After delays ranging from minutes to days the subjects were tested to determine how much they had retained. The tests were quite sensitive–they did not merely ask "What do you remember?" but often used true/false or multiple choice questions, in which even a vague memory of the material would allow selection of the correct choice. Because experiments by many different experimenters were summarized and analyzed, the results of the analysis are fairly robust; they are insensitive to fine details or specific conditions of one or another experiment. Finally, the amount remembered was divided by the time allotted to memorization to determine the number of bits remembered per second.

The remarkable result of this work was that *human beings remembered very nearly two bits per second under all the experimental conditions.* Visual, verbal, musical, or whatever–two bits per second. Continued over a lifetime, this rate of memorization would produce somewhat over 10^9 bits, or a few hundred megabytes. This estimate is surprisingly identical to Turing's estimate. But the issue is far from being resolved. I will only mention an estimate nearer to the estimate of von Neumann. Moravec (10) recently tried to compare computer hardware and the brain. He estimated the memory capacity as 100 million megabytes, which are about 10^{15} bits.

VII. COMPUTATIONAL LEARNING THEORY

Complexity issues are dealt with in the areas computability theory, complexity theory, theory of inductive inference, and computational learning

theory. Computability theory investigates what can be computed, the theory of inductive inference what can be learned at all. They are historically prior to and part of their polynomially-obsessed younger counterparts. In fact, Turing founded computability theory and made the major contribution.

In this section I will concentrate on computational learning theory, because it fulfills von Neumann's requirement to investigate the space and the number of steps to learn a problem. The following review is based on the survey of Angluin (1). He defines the goals of the field as:

Give a rigorous computationally detailed and plausible account of how learning can be done.

These goals are far from being achieved. There is even not an agreement on a precise definition of learning. So far the emphasis has been on inductive learning and particular PAC (probably approximately correct learning) introduced by Valiant 1984 (20). In this framework the learner gets samples that are classified according to a function from a certain class. The aim of the learner is to find an approximation of the function with high probability. We demand the learner to be able to learn the concept given any arbitrary approximation ratio, probability of success or distribution of the samples.

More precisely:
Algorithm A PAC-identifies concepts from C in terms of the hypothesis space H if and only if for every distribution D and every concept $c \in C$, for all positive numbers ϵ and δ and access to the example oracle, it eventually halts and outputs a concept $h \in H$ that with probability at least $1 - \delta$ and error $D(c\Delta h) < \epsilon$, where $c\Delta h$ is the symmetric difference between the subsets of X characterizing the concepts c and h. The model was further extended to treat noise (misclassified samples). There have been lots of interesting results achieved. But until today many problems are open. I just mention the problem if distributed normal forms DNF in Boolean space are PAC-learnable in polynomial time. This result supports von Neumann's feeling that simple learning mechanisms lead to the curse of exponential enumeration.

VIII. How to get common sense into a machine

Turing's idea of creating first a child machine was reinvented by John McCarthy in 1999 (8). He wrote an essay on an artificial child brain as a step towards creating human like intelligence. He writes in the abstract:

"The innate mental structure that equips a child to interact successfully with the world includes more than universal grammar. The world itself has structures, and nature has evolved brains with ways of recognizing them and representing information about them. For example, objects continue to exist when not being perceived, and children (and dogs) are very likely "designed" to interpret sensory inputs in terms of such persistent objects. Moreover, objects usually move continuously, passing through intermediate points, and perceiving motion that way may also be innate. What a child learns about the world is based on its innate mental structure."

Thus McCarthy notices in contrast to Turing that the innate mental structure is not a sheet of blank paper, but it is very complicated shaped by evolution. McCarthy tries to design adequate mental structures including a language of thought. "This design stance applies to designing robots, but we also hope it will help understand universal human mental structures. We consider what structures would be useful how the innateness of a few of the structures might be tested experimentally in humans and animals." The proposal was never finished and remained a paper proposal. Therefore the issue of creating a suitable child machine is still unsolved. At this time nobody seems working on this problem.

I also tried to combine evolution and learning for automatic programming (14). But good results have been obtained only in the separate domains, neural networks (25) and optimization by simulating evolution (11).

The other approach to machine intelligence is still pursued in a big project. This means coding all the necessary common sense knowledge into some computer understandable description. We remind the reader, that this method was considered as too inefficient, both by Turing and von Neumann. Von Neumann even doubted if this method would work at all. The project was started in 1984 with the name *Cyc*, the goal of which was to specify in a well-designed language common sense knowledge. Cyc is an artificial intelligence project that attempts to assemble a comprehensive ontology and database of everyday common sense knowledge, with the goal of enabling AI applications to perform human-like reasoning. The original knowledge base is proprietary, but a smaller version of the knowledge base, intended to establish a common vocabulary for automatic reasoning, was released as OpenCyc under an open source license.

Typical pieces of knowledge represented in the database are "Every tree is a plant" and "Plants die eventually".

When asked whether trees die, the inference engine can draw the obvious conclusion and answer the question correctly. The Knowledge Base (KB) contains over a million human-defined assertions, rules or common sense ideas. These are formulated in the language CycL, which is based on predicate calculus and has a syntax similar to that of the Lisp programming language.

Much of the current work on the Cyc project continues to be knowledge engineering, representing facts about the world by hand, and implementing efficient inference mechanisms on that knowledge. Increasingly, however, work at Cycorp involves giving the Cyc system the ability to communicate with end users in natural language, and to assist with the knowledge formation process via machine learning. Currently the knowledge base consists of

- 3.2 million assertions (facts and rules)
- 280,000 concepts
- 12,000 concept-interrelating predicates

I cannot evaluate Cyc in detail, but despite its huge effort the success is still uncertain. Up to now Cyc has not been successfully be used for any broad AI application.

IX. The problem of initiative or meta-learning

From all the research in this very challenging area I will only review the work done in connection with neural networks. Even today learning in neural networks is typically done "from scratch" without using previous knowledge. This follows from the fact that learning begins from initially random connection weights. A first step to using previous knowledge was cascade correlation (CC) (2). It creates a network topology by recruiting new hidden units into a feed-forward network in order to reduce the error.

This algorithm has been extended to knowledge-based cascade correlation (KBCC) which recruits whole sub-networks that it has already learned, in addition to the untrained hidden units recruited by CC (16). KBCC trains connection weights to the inputs of its existing sub-networks to determine whether there outputs correlate well with the network's error on the problem it is currently learning. The previously learned networks compete with each other and with conventional untrained candidate hidden units to be recruited into the target network learning the current problem.

The general idea sounds convincing, but for an implementation a number of difficult decisions have to be made. If, for instance, all previously learned sub-networks compete with each other, the learning will slow down with the number of problems to be learned. The current results of KBCC are still very preliminary. In

(16) an evaluation is done using only two problems. In the first setting it is evaluated whether KBCC can find and use its relevant knowledge in the solution of a new problem similar to the first one. In the second setting it is investigated whether KBCC can find and combine knowledge of components to learn a new, more complex problem comprised of these components. The results indicate that it is worthwhile to develop KBCC further, but it is unclear how KBCC would perform on larger problems. Thus Turing's initiative problem remains unsolved.

X. Probabilistic logic

The theory of probabilistic logic has been fully developed in the last 20 years. Uttley invented a conditional probability computer as early as 1958 (19). The major drawback of his design was that in order to classify an input of n binary items, the number of neurons had to be exponential 2^n. It took quite a while to solve this problem and to see the connection of probabilistic logic to probability theory. A very popular instance of probabilistic logic are Bayesian networks.

The problem of the exponential explosion has been solved in the 80's. For singly connected Bayesian networks exact inference is possible in one sweep of Pearl's belief propagation algorithm (15). A very interesting extension for incomplete data is done by the maximum entropy principle (3). This theory can be seen as a realization of von Neumann's prophesy.

Probabilistic logic is now used in many fields. To give just one example, I have applied Bayesian networks to population based global optimization (13).

XI. Complication and complexity

The complication problem formulated by von Neumann has still not been formulated in a precise scientific manner. For the reader I restate the problem: "It is possible that the connection pattern of the visual brain itself is the simplest logical expression or definition of this principle (visual analogy)". In this section I will just mention important contributions to the solution of this problem which might later lead to a scientific theory. Nearest to the thinking of von Neumann comes *algorithmic complexity* (also known as descriptive complexity, Kolmogorov-Chaitin complexity) (5).

The Kolmogorov complexity of an object such as a piece of text is a measure of the computational resources needed to describe the object. To define Kolmogorov complexity, we must first specify a description language for strings. Such a description language can be based on a programming language such as Lisp, C++, or Java virtual machine byte-code. If P is a program

which outputs a string x, then P is a description of x. The length of the description is just the length of P as a character string. In determining the length of P, the lengths of any subroutines used in P must be accounted for. The length of any integer constant n which occurs in the program P is the number of bits required to represent n, that is (roughly) log2n. We could alternatively choose an encoding for Turing machines (TM), where an encoding is a function which associates to each TM M a bit-string $<M>$. If M is a TM which on input w outputs string x, then the concatenated string $<M>, w$ is a description of x. For theoretical analysis, this approach is more suited for constructing detailed formal proofs and is generally preferred in the research literature. Note that Kolmogorov complexity is valid for a single string only.

We cite some important results. Let $K(s)$ denote the complexity of string s. Obviously $K(s)$ cannot be too much larger than the string itself.

$$K(s) \leq |s| + c$$

A string s is compressible by c if it has a description whose length does not exceed $|s| - c$. This is equivalent to saying $K(s) \leq |s| - c$. Otherwise s is incompressible by c. A string incompressible by one is said to be simply incompressible; by the pigeonhole principle, incompressible strings must exist, since there are 2^n bit strings of length n but only 2^{n-1} shorter strings, that is strings of length $n - 1$. For the same reason, "most" strings are complex in the sense that they cannot be significantly compressed: $K(s)$ is not much smaller than $|s|$, the length of s in bits. To make this precise, fix a value of n. There are 2^n bit strings of length n. The uniform probability distribution on the space of these bit strings assigns to each string of length exactly n equal weight 2^{-n}.

Theorem 1: With the uniform probability distribution on the space of bit strings of length n, the probability that a string is incompressible by c is at least $1 - 2^{-c+1} + 2^{-n}$.

This means that "most" strings cannot be compressed. Thus in this limited domain (just a single string) this result is almost the opposite to the conjecture of von Neumann. Kolmogorov complexity has been extended to sets of strings and functions. In (21) a generalization of Kolmogorov complexity is described which unifies some of the most important principles of machine learning, like the minimum description length MDL, Occam's razor and Shannon's entropy. This topic is far too difficult to be discussed here.

XII. CONCLUSION AND OUTLOOK

I hope the reader is as astonished as I was when reading the papers of Turing and von Neumann. In my opinion they have discussed all aspects and components which seem necessary to develop human like artificial intelligence. Both researchers had no doubts that any problem which can be precisely formulated can also be programmed. Turing concentrated his design for machine intelligence on the construction of a child machine and learning. Von Neumann had doubts that it will be possible to construct machine intelligence by programming or by learning. It leads to the curse of infinite enumeration. Therefore he asked the bold question if it is possible that automata could develop to higher complexity without too much human intervention. He succeeded to construct a self-reproducing automata, but did not have time to investigate the next step, namely simulating evolution to breed automata of higher complexity.

Turing identified the following major problems on the road to human like machine intelligence
- What are the minimal requirements for a child machine to allow efficient learning?
- How can learning be made more efficient than using punishment and reward?
- What has to be done that the machine actively learns using initiative?

Von Neumann formulated the following problems
- The lack of a logical theory of automata
- The limited complexity of artificial automata
- A rigorous concept of what constitutes "complication"

From these problems only the logical theory is solved, the other five are still open. But for the construction of complex automata the theoretical results are often negative if we require that the "chains of reasoning" (von Neumann) are finite, e.g. polynomial. A major achievement has been the precise formulation of probabilistic logic. Despite a number of efforts there has been no progress in extending von Neumann's self-reproducing automata with some evolution mechanism so that they become substantial more complex.

In the sixty years after the ground braking work of Turing and von Neumann a lot of impressive systems have been built which solve precisely defined problems. These are too many to cite here. But there is no system in sight which comes near to passing the Turing test. In current competitions the machine is identified after a few questions. What might be the reason for the slow progress? The simple answer is that there has been no substantial progress to solve the remaining five problems identified by Turing and von Neumann.

A machine with human like intelligence needs *common sense* reasoning, the sort of reasoning we would expect a child easy to do. The relative paucity of results in

this field does not reflect the considerable effort that has been expended, starting with McCarthy's paper "Programs with Common Sense" (6)[3]. Forty years after the first paper McCarthy notices that the knowledge needed to solve a commonsense reasoning problem is typically much more extensive and general than the knowledge needed to solve difficult scientific problems in mathematics or physics (7). There the knowledge is bounded. In contrast, there are no a priori limitations to the facts that are needed to solve commonsense problems: the given information may be incomplete; one may have to use approximate concepts and approximate theories; and one will need some ability to reflect upon one's own reasoning process.

What recommendations I can give to young scientists working in this area? First, try to make contributions to the open problems before trying a general architecture. Most important topics are higher learning methods like meta-learning or even transfer learning, Turing called this providing the machine with *initiative*. Second, von Neumann's proposal to start with self-reproducing automata is also worthwhile to investigate further. But here I am very sceptical that this way will ever lead to human like intelligence. But it will certainly give new insights to biological problems.

<div align="center">References</div>

[1] D. Angluin. Computational learning theory: Survey and selected biography. In *Proceedings of the 24th ACM Symposium on the Theory of Computing*, pages 351–369, New York, 1992. ACM Press.

[2] S. Fahlman and C. Lebiere. The cascade-correlation learning algorithm. In D.S. Touretzky, editor, *Advances in Neural Information Processingg*, volume 2, pages 524–532, San Mateo, 1990. Morgan Kaufman.

[3] E.T. Jaynes. Information theory and statistical mechanics. *Phys. Rev*, 6:620–643, 1957.

[4] Th. K. Landauer. Estimates of the quantity of learned information in long-term memory. *Cognitive Science*, 10:477–493, 1986.

[5] Ming Li and P. Vitanyi. *An Introduction to Kolmogorov Complexity and its Application*. Springer, Heidelberg, 2002.

[6] J. McCarthy. Programs with common sense. In *Mechanisation of Thought Processes*, pages 75–84. Her Majesty's Stationery Office, London, 1959.

[7] J. McCarthy. From here to human-level intelligence. In *Proceedings 5th Conference on Knowledge Representation and Reasoning*, pages 640–646. Morgan Kaufmann, San Mateo, 1996.

[8] John McCarthy. The well-designed child. Technical report, Stanford University, 1999.

[9] W.S. McCulloch and W. Pitts. A logical calculus of the ideas immanent un nervous activity. *Bull. of Mathematical Biophysics*, 5:115–137, 1943.

[10] H. Moravec. When will computer hardware match the human brain? *Journal of Evolution and Technology*, 1:1–14, 1998.

[11] H. Mühlenbein. Evolution in time and space - the parallel genetic algorithm. In G. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 316–337. Morgan Kaufmann, San Mateo, 1991.

[12] H. Mühlenbein. Towards a theory of organisms and evolving automata. In A. Menon, editor, *Frontiers of Evolutionary Computation*, pages 1–36. Kluwer Academic Publishers, Boston, 2004.

[13] H. Mühlenbein and R. Höns. The estimation of distributions and the minimum relative entropy principle. *Evolutionary Computation*, 13(1):1–27, 2005.

[14] H. Mühlenbein and J. Kindermann. The dynamics of evolution and learning - towards genetic neural networks. In R. Pfeiffer, editor, *Connectionism in Perspectives*, pages 173–198. North-Holland, 1989.

[15] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman, San Mateo, 1988.

[16] T. R. Shultz and F. Rivest. Knowledge-based cascade-correlation: Using knowledge to speed learning. *Connection Science*, 13:1–30, 2002.

[17] A.M. Turing. Computing machinery and intelligence. *Mind*, 59:433–460, 1950.

[18] A.M. Turing. Intelligent machinery. In B. Meltzer and D. Michie, editors, *Machine Intelligence 6*, pages 3–23. Oxford University Press, Oxford, 1969.

[19] A. M. Uttley. Conditional probability computing in a nervous system. In *Mechanisation of Thought Processes*, pages 119–152. Her Majesty's Stationery Office, London, 1959.

[20] L. G. Valiant. A theory of the learnable. *C. ACM*, 27:1134–1142, 1984.

[21] N. K. Vereshchagin and P. Vitanyi. Kolmogogorov's structure function and model selection. *IEEE Transactions on Information Theory*, 50:3265–3290, 2004.

[22] J. von Neumann. The general and logical theory of automata. In *The world of mathematics*, pages 2070–2101. Simon and Schuster, New York, 1954.

[23] J. von Neumann. Probabilistic logics and the synthesis of reliable organs from unreliable components. In *Annals of Mathematics Studies 34*, pages 43–99. Princeton University Press, 1956.

[24] J. von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, Urbana, 1966.

[25] Byoung-Tak Zhang and H. Mühlenbein. Balancing accuracy and parsimony in genetic programming. *Evolutionary Computation*, 3:17–38, 1995.

[3] In the discussion of the paper Bar-Hillel said: "Dr. McCarthy's paper belongs in the Journal of Half-Baked Ideas...The gap between McCarthy's general programme and its execution seems to me so enormous that much more has to be done to persuade me that even the first step in bridging this gap has already been taken