1956-1966 How Did It All Begin? - Issues Then and Now

Marvin Minsky

MIT Media Lab and MIT AI Lab
Professor Emeritus, Media Arts and Sciences Professor of E.E. and C.S., M.I.T
minsky@media.mit.edu

Extended Abstract

Many computer programs today show skills that appear to rival those of outstanding human consultants. However, while each such program does certain things well, it is helpless at doing anything else. Why do our present-day programs lack the versatility and resourcefulness that a typical person shows? Clearly, those programs are deficient in both commonsense knowledge and commonsense reasoning. I'll argue that this has happened because the field of AI has evolved in a backwards direction, as compared with how a typical person develops-and that this is because our AI programmers have not appreciated the importance of making their system able to use more 'reflective' ways to think.

We can see this backwards trend in the earliest years. Consider the following list of AI accomplishments.

1957 Arthur Samuel: A machine that plays master-level Checkers.
1957 Newell, Shaw and Simon. Proving theorems in Propositional Logic.
1960 Herbert Gelernter: Proving theorems in Euclidean Geometry
1960 James Slagle: Symbolic Integral Calculus
1963 Lawrence G. Roberts: 3-D Visual Perception
1964 Thomas G. Evans: Solving Geometry Analogy problems.
1965 Daniel Bobrow: Solving word problems in Algebra
1969 C. Engelman, W. A. Martin and J. Moses: the MACSYMA project.
1969 Minsky, Papert, et al: A robot builds structures with wooden blocks.
1970 Patrick Winston: A robot that learns to recognize such structures.
1970 Terry Winograd: A program that understands many sentences.
1972 Gerald Sussman: A program that recognizes some bugs in programs.
1974 Eugene Charniak: A program that understands a few simple stories

Although there are many exceptions to this, one can discern a trend in which the early programs made progress at 'expert' tasks-whereas the later programs attempted to do things that typical four-year-olds do. In retrospect, we can clearly see that those 'more advanced' skills were easier because they required less commonsense knowledge and reasoning. The situation is still the same today: no visual program can recognize the objects in a typical room, or answer simple-seeming questions about the stories in a typical first-grade storybook.

In subsequent years most AI researchers aimed to discover some single problem-solving technique could keep extending itself. Consequently the field of AI divided

C. Freksa, M. Kohlhase, and K. Schill (Eds.): KI 2006, LNAI 4314, pp. 437–438, 2007.

itself into such specialties as Rule-Based Systems, Artificial Neural Networks, Statistical Inference Systems, Formal Mathematical Logic, Genetic Programs, and so on. Each of these were effective in certain types of situations, but never became very competent in other kinds of realms of domains.

It seems to me that what went wrong was that too many researchers became advocates and too few attempted to discover the limitations of their favorite method. Instead, most such researchers publish only instances in which their favorite scheme solves some particular problem, but the literature shows little discussion or classification of the realms in which each such system fails.

For example, in 1969 Seymour Papert and I published a book that showed some serious limitations of certain three-layer non-reentrant neural networks-but most researchers in that field have wrongly assumed that those limitations would not hold for such networks with more layers. However, so far as I can see, almost all of our theorems still apply, in the sense that the size of the networks and their coefficients still grow exponentially with the scale of the problem. The basic problem is that such networks compute only non-recursive functions. In other words, they simply cannot do any 'reflective thinking' about why their recent activities failed or succeeded-and this limited the effectiveness of what they could learn from experience. An attempt to describe a more reflective system appeared in a 1960 paper by Newell, Shaw, and Simon, but I have seen no later references to that proposed approach.

My lecture will describe some ideas about how we could build more resourceful machines, by designing system that can learn when and how to switch between many different ways to think. More details about these ideas are discussed on my website at http://web.media.mit.edu/~minsky/E8/eb8.html. If there is time, I will also discuss the early years of working with Warren McCulloch and other pioneers.