Steps Towards Self-Organizing Intelligence

Michiel de Jong

October 19, 2001

Abstract

This paper proposes a systematic approach to the problem of building artificially intelligent computer systems. The key idea is to use a top-down approach in the lower layer, and a bottom-up approach in a higher layer. That way, known solutions for specific easy subproblems can be used. At the same time they can be combined into a larger system in such a way that the result is robust. The significance in this context of recent and future developments in internet technology are also discussed.

1 Introduction

Artificial Intelligence (AI) is the field of research that is concerned with trying to build computer programs that are as intelligent as humans. This paper proposes a specific approach to AI, that can be called "self-organizing intelligence". The term self-organizing emphasizes the feature that the system organizes itself, and is not explicitly synthesized by human engineers. This approach combines the best aspects of the two approaches that have dominated AI research through the years, namely the bottom-up approach and the top-down approach. This paper will also try to predict the possible future roles of the internet in AI.

2 A short history of Artificial Intelligence

The ultimate goal of Artificial Intelligence (AI) is to build an artificial system that has at least the level of intelligence humans have. Forty years ago, in a paper called "Steps Towards Artificial Intelligence" [9], Minsky concluded that important issues in artificial intelligence were search, pattern recognition, learning, planning, and induction. Since then, a lot of progress has been made on these separate tasks. Separate systems and solutions have been put forward, each of which is concerned with one of these five tasks.

Much of the research in AI can be categorized as either top-down AI or bottom-up AI. Initially, most AI research was done from a top-down perspective. This was partly because, like computer scientists in general, most AI researchers had a background in mathematics. In mathematics, specific propositions are derived by combining more general axioms and transition rules. That is a top-down approach, which starts at the general problem as a whole, and then works towards more specific subproblems. This approach can be said to have failed, because it can lead to very brittle solutions. If a top-down AI system works well

under the circumstances for which it has been designed, then it will probably fail miserably under slightly different conditions.

In reaction to the inability of top-down AI to get convincing results, bottom-up AI became very popular. Bottom-up AI starts with small and easy problems, and tries to grow more intelligent from there, much like a child growing up and facing ever more challenging tasks. The reason why bottom-up AI succeeded in escaping the brittleness problems of top-down AI, is that these systems learn their abilities themselves, and are not taught how to do them by anybody. So if a bottom-up AI system has to do a slightly different task, it will probably have the ability to learn that task too. The strength of bottom-up AI is for a large part in the robustness of its solutions, as is argued in for instance [2, 6, 9, 1]. There has however been a regress of enthousiasm about bottom-up AI too. During the last few years, people have realized that letting the system learn everything by itself has its limitations. As the problems get harder, bottom-up AI systems will need more and more time to build up the expertise required for these problems. Anything you don't put in, has to be found by random search. Bottom-up AI is more about finding easy problems than about finding good solutions for hard problems.

3 A bottom-up society of top-down agents

To unite the strengths of top-down and bottom-up AI, we propose the use of manmade building-blocks (in the spirit of top-down AI), and to organize these in a self-organizing structure (as is done in bottom-up AI). That would overcome the brittleness of top-down AI systems, because the self-organizing society of agents has enough redundancy and robustness to overcome violations of assumptions about a problem. At the same time, it would overcome the lack of efficiency portrayed by bottom-up AI systems, because it uses man-made ingredients that are known to be far better than anything a bottom-up agent would come up with by random search.

4 AI's ultimate goal: the Turing Test

The ultimate goal of AI can be formulated by means of an abstraction that has been called the Turing Test [11]. In this test, the computer that is a candidate for being intelligent is placed in a room, and in another room a human is placed. In a third room, a judge is placed, who can pose questions to "A" or to "B". The judge doesn't know which one of these is human, and which one is the computer. An errand runner walks between the three rooms, to deliver the questions and the answers. He also types the questions into the keyboard of the computer, and reads the answers from its screen. If the judge can't make out which answers come from the human and which ones are generated by the computer, the computer is said to be "artificially intelligent".

5 Multi-layer Methods and the Optimal Optimizer

The approach we propose for working towards such an artificially intelligent computer system, is through a bottom-up society of top-down agents. The agents can be very good at a specific task, because they are the product of years of engineering. For instance, there could be an agent applying the Lin-Kernighan algorithm to Traveling Salesman problems. This algorithm is the best known heuristic for problems of this type. There is no way such a complex method could have been evolved from scratch by self-organization. It is much too specific for that. On the higher level though, deciding which agent will do which task should be a self-organizing process. Otherwise, unforeseen circumstances could very easily expose errors in the design that cannot be overcome by a rigid top-down system, and the system would come to show brittleness. Therefore, the society of agents as a whole should be bottom-up.

So the approach we propose has two layers: a higher society-layer, and a lower agent-layer. Therefore, it qualifies as a multi-layer method. In a multi-layer method, a higher layer controls the submethods on the lower layer. Such methods abound in computer science [5, 4, 10, 1].

In [9], Minsky states that a specific submethod is useful if at the one hand, it is really good at some subclass of a problem, and on the other hand this subclass is easily recognizable. If it takes a lot of effort to determine whether a given method will or will not be applicable, then its being fast is no use after all, because too much time is spent before its application. We conjecture that the optimal way of approximating an optimization problem is by looking for specific features of the problem, and then deciding which submethod to invoke, based on these features. Recursively, this submethod will start looking for further characteristics, and invokes an even more specific method, etcetera. If the right submethods are connected to the right decisions, this would lead to the optimal optimizer. That is, the method that is optimally fast at general optimization problems. This can probably be proved formally without much effort. It would be the speed-prior variant of the Minimal Description Length Principle.

6 A methodology for experimental computer science

To create an artificially intelligent computer system, we would have to think up lots of submethods for it, and define for each submethod under which circumstances we expect it to be applicable. The former part is done feverishly in experimental computer science, but the latter is often not looked after very carefully.

If the field is to make any significant progress, the methodology of experimental computer science will have to change. Most importantly, it will have to become more centralized. Currently, people report results of comparing their new algorithm to an algorithm of Brand X, on some problem class, concluding that their algorithm is superior. But having all these independent experiments stack up in the literature, doesn't really bring us much further in the whole picture of our quest for the optimal optimizer. The problem is that the individual

experiments are never entirely comparable to each other. Therefore, we propose that there should be a central repository, keeping track of which algorithm is good for which problem, and judging this independently. If this could work out, this central repository would already constitute our best current approximation of the optimal optimizer.

7 The internet as a new circumstance

Early analyses of the goals and tasks of artificial intelligence, did not consider the internet as a factor yet. This section points out three ways in which the internet can change artificial intelligence research.

7.1 Utilizing remote idle-time

One big possible advantage of the internet is the availability of idle-time. Idle-time is the time spent by a computer's processor without doing anything. When the user is typing text, the computer uses its processor only for a fraction of a second after each key stroke, and then goes back into slumber. The processor time that is not used by any of the processes that the user has running is called idle-time. This idle-time can in principle be use for anything without slowing down the responsiveness of the computer to the main user. In January 1996, there were 9.5 million computers connected to the internet, 1.7 million of which were turned on and functioning simultaneously. Most computers are utilized for only at most 20% of the time, so that means that four years ago already, there was a potential of about half a million CPUs of computer power if one could only get all the other computers on the internet to collaborate. Examples of this principle are the SETI@HOME project and the prime factorization of large numbers here at CWI.

Like with parallel computers, remote idle-time is hardly ever used in practice, despite the obvious potential gains. We think that this will change over time, as mobile agent technology [8, 7, 3] becomes more popular. Mobile agent technology facilitates the use of remote computer time greatly. A programmer can easily make a mobile agent migrate to another computer and continue its business there.

7.2 The internet as a single intelligence

Having access to all the idle-time, software, and data on the internet would boost the possibilities for building anything artificially intelligent enormously. The connection of such a system to the Turing test is already evolving in the ever more clever search engines. For instance, MetaCrawler acts like a multi-layer method, that uses other search engines as its submethods. Also, CiteSeer uses a smart trick, namely studying how scientific papers reference each other, to be an extremely useful search engine specifically tailored for finding scientific papers.

As search engines get better and better, and maybe one day will have a natural language interface, they get closer and closer to a system that would pass the Turing test. Already, for a lot of questions, it's more productive to "ask" an internet search engine than to ask a human.

8 Conclusions

This paper proposed a way to approach the goals of artificial intelligence. This consists of a combination of top-down design at the lower level, and bottom-up design at a higher level. Also, the implications of the future of the internet were discussed.

References

[1] Eric B. Baum. Manifesto for an evolutionary economics of intelligence. In *Neural Networks and Machine Learning*. Springer Verlag, 1998.

[2] Rodney A. Brooks. Intelligence without reason. In *Proc. IJCAI 1991*, 1991.

[3] Gianni Di Caro and Marco Dorigo. Mobile agents for adaptive routing. In *Proceedings of the thirty-first Hawaii International Conference on Systems*, January 1998.

[4] T. J. Chang, N. Meade, and J. E. Beasley. Heuristics for Cardinality Constrained Portfolio Optimization. Technical report, The Management School, Imperial College, London SW7 2AZ, England, May 1998.

[5] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting, 1998. Available by anonymous download from http://www-stat.stanford.edu/~jhf/ftp/boost.ps.

[6] John H. Holland. *Escaping Brittleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems*, pages 593–623. publisher, 1986.

[7] Kwindla Hultman Kramer, Nelson Minar, and Pattie Maes. Tutorial: Mobile software agents for dynamic routing. *Mobile Computing and Communications Review*, 3, 1999.

[8] Nelson Minar, Kwindla Hultman Kramer, and Pattie Maes. *Cooperating Mobile Agents for Dynamic Network Routing*, chapter 12. Springer-Verlag, 1999.

[9] Marvin Minsky. Steps towards artificial intelligence. In E.A. Feigenbaum and J. Feldman, editors, *Computers and Thought*, pages 406–450, New York, 1963. McGraw-Hill.

[10] J. Schmidhuber. *A general method for incremental self-improvement and multiagent learning*, chapter 3, pages 81–123. Scientific Publ. Co., Singapore, 1999.

[11] A.M. Turing. Computing machinery and intelligence. In E.A. Feigenbaum and J. Feldman, editors, *Computers and Thought*, pages 11–35. McGraw-Hill, New York, 1963.