Psychology and Intelligent Machines

Ovais Ahmad Khan (s1105)
ovais.khan at nu.edu.pk

Table of Contents

Introduction

It is widely perceived that machines cannot even come close to acting like a person. You must have overheard some of the comments like this:

> *"Of course machines can do useful things. We can load them with programs that go through the motions of adding sums of numbers up, or assembling parts in factories. But those are just dull mechanical functions – and there's no way machines could have genuine feelings."*

All of us will definitely agree that machines are far more superior when it comes to logical things. But for a moment think about love, emotions, sufferings and lot's of other things that we as a human experience. Think about our consciousness and common sense. How hard is it to explain how am I writing this assignment at this moment.

Psychology has got answers to and theories for many of these phenomenons. We can definitely use the knowledge gained by centuries of research of psychologists to make a machine that can do certain things like us and this is the topic of this report.

The basic problem is the difference in the goal of the two – Psychology and Artificial Intelligence. First, if AI is to mimic the human-like methods, then it must also imitate error. Secondly, there is the question of efficiency versus plausibility. A deep trend is driving away AI from "human-like" methods to compute-intensive methods. Also, the cognitive theories must be able to predict the difficulty of various tasks in humans.

Most of the research is being conducted to make *intelligent agents* – software programs that can perform functions on the behalf of its owner and are able to take decisions on their own as the situation demands without requiring the intervention of the owner. Defining intelligence would be futile. We just want to synthesize intelligent systems with whichever available means, rather then to model human behavior.

Machine Learning and *Common Sense knowledge and Reasoning are* the fields in which psychology (especially Cognitive Psychology) has played a very prominent role, and for this we mostly tried to mimic human behavior. Let's quickly have an overview of some of the ways in which we are trying to build intelligent machines.

SAIL and Dav – Developmental Robots

John Weng, a robotics expert at Michigan State University, is teaching a robot to learn like a child -- to obey spoken commands, trundle down a hall, and find and pick up toys with its mechanical hand.

Named SAIL (Self-organizing Autonomous Incremental Learner), Weng calls his machine a *developmental robot*, because, unlike most traditional robots, it develops its new abilities through practice, gaining skill with each training session. When a kid first opens its eyes, it doesn't understand what it is seeing, this comes later. The same goes for his AI robot SAIL.

What is a Developmental Learning

For humans, the developmental algorithm is in the genes. It starts to run at the conception time of each human individual. This algorithm is responsible for whatever can happen through the entire life span of that individual. For machines, the developmental algorithm starts to run at the ``birth" time of the machine, enabling the machine to develop its cognitive and behavioral skills through direct interactions with its environment using its sensors and effectors. For grounded development for truly machine understanding, the environment must be the physical world which includes the human teachers and the robot itself.

The concept of developmental algorithm does not mean just to make machines to growth from small to big and from simple to complex. It must enable the machine to learn new tasks that a human programmer does not know about at the time of programming. This implies that the representation of any task that the robot learns must be generated by the robot itself, a well known holy grail in AI and fundamental for machine understanding.

The basic nature of developmental learning plays a central role in enabling a human being to incrementally scale his or her level of intelligence from ground up. In order to scale up the machine's capability to understand what happens around it, the learning mechanism embedded in a developmental algorithm must perform systematic self-organization, according to what it sensed, what it did, action imposed by the human when necessary, the reward it received from the humans, and the context. As a fundamental requirement of scaling up, the robot must develop a value system.

How SAIL learns:

> *"Humans mentally raise the developmental robot by interacting with it. Human trainers teach robots through verbal, gestural or written commands in much the same way as parents teach their children."* -- Weng

SAIL acquires its smarts in two ways: The first is through supervised learning under the direct control of a human teacher. Then comes reinforcement learning, in which the trainer lets the robot operate on its own but rewards it for successful action and penalizes it for failure.

In supervised learning, for example, the trainer steers SAIL down a corridor by pushing touch sensors on its shoulders. To train the baby robot to get around, we will have to take it for a walk. After a few practice sessions, the machine is free to go on it's own. SAIL needs only one lesson to learn to move in a straight line, but 10 sessions to get the hang of going around corners on its own.

In another type of lesson, the human trainer speaks an order, such as "go left," "arm up" or "open hand", then makes the robot perform the action by pushing one of the 32 control sensors on its body. The robot associates what it hears with the correct action. According to Weng, after 15 minutes' training, SAIL could follow such commands correctly 90 percent of the time, he said.

The robot next attends an advanced class of reinforcement learning to strengthen its newfound skills. The trainer lets the robot explore the world on its own, but encourages and discourages its actions by pressing its 'good' or 'bad' buttons. Alternatively, a trainer can say ‘Good’ or ‘Bad’. The 'good' and 'bad' commands speed up the learning process. *“Mind and intelligence emerge gradually from such interactions”*, Weng said.

How long will all this take?

Although, this learning process takes time, Weng explains that it does not necessarily encompass 21 years as it would in humans, who are limited by biological constraints. Developmental robots can 'live' with us and become smarter autonomously, under our human supervision.

How long can a living machine live? When the hardware of a living machine is warn or broken, the developmental algorithm with its learned knowledge can be downloaded from the machine and uploaded to a new physical machine. Therefore, unlike a biological brain, a developmental algorithm can run as long as we humans like.

What’s next – Dav

Development is the second such effort after SAIL. It had got a more human-looking body.

More information about Dav and SAIL can be obtained from http://www.cse.msu.edu/~weng/research/LM.html.

Cyc & OpenCyc - knowledge base and commonsense reasoning engine

Cyc® pronounced as "psych" has been developed by Cycorp, Inc., based in Austin, Texas. OpenCyc is the open source version of Cyc. It is an immense multi-contextual knowledge base having all of the "trivial" facts of common sense and an efficient inference engine. The knowledge base is built upon a core of over 1,000,000 hand-entered assertions (or "rules") designed to capture a large portion of what we normally consider consensus knowledge about the world. Cyc has got many basic concepts, assertions, rules, inference engine and it also includes dialog tools that can easily extract information from domain experts.

What is it really?

Cyc is the complement of an encyclopedia. The aim is that one day Cyc ought to contain enough commonsense knowledge to support natural language understanding capabilities that enable it to read through and assimilate any encyclopedia article, that is, to be able to answer the sorts of questions that you or I could after having just read any article, questions that neither you nor I nor Cyc could be expected to answer beforehand.

Where they can be used

Cyc or OpenCyc can be used as the basis of a wide variety of intelligent applications such as speech understanding, database integration, rapid development of an ontology in a vertical area, email prioritizing, routing, summarization, and annotating, to name just a few. They can tell us about:

- Cyc can find the match between a user's query for "pictures of strong, adventurous people" and an image whose caption reads simply "a man climbing a cliff."
- Cyc can notice if an annual salary and an hourly salary are inadvertently being added together in a spreadsheet.
- Cyc can combine information from multiple databases to guess which physicians in practice together had been classmates in medical school.
- When someone searches for "Bolivia" on the Web, Cyc knows not to offer a follow-up question like "Where can I get free Bolivia online?"

What's in OpenCyc

- 6,000 concepts: an upper ontology for all of human consensus reality.
- 60,000 assertions *about* the 6,000 concepts, interrelating them, constraining them, in effect (partially) defining them.
- A compiled version of the Cyc Inference Engine and the Cyc Knowledge Base Browser.
- A suite of "RKF" tools for rapidly extracting knowledge from a domain expert (e.g., a physician or oil drilling specialist), tools which operate by carrying on a clarification dialogue with that individual; hence: tools for answering questions via English dialogue.

- Documentation and self-paced learning materials to help users achieve a basic- to intermediate-level understanding of the issues of knowledge representation and application development using Cyc.
- A specification of CycL, the language in which Cyc (and hence OpenCyc) is written. There are CycL-to-Lisp, CycL-to-C, etc. translators.
- A specification of the Cyc API, by calling which a programmer can build an OpenCyc application with very little familiarity with CycL or with the OpenCyc KB.
- The ability to import and export CycML files.
- A few sample programs that demonstrate use of the Cyc API for application development.

How it works

It all begins with assembling a collection of partial solutions to the various difficult problems Cyc has to handle, and add new tools as required. That is, for a number of problems (time, causality, inference, user interface, and so on), there aren't any known general-purpose, simple, efficient solutions, but we can make do with a set of modules that enable us to easily handle the most common cases.

The bulk of the effort is currently devoted to identifying, formalizing, and entering microtheories of various topics (such as shopping, containers, emotions). We follow a process that begins with a statement, in English, of the microtheory. On the way to our goal, an axiomatization of the microtheory, we identify and make precise those Cyc concepts necessary to state the knowledge in axiomatic form. To test that the topic has been adequately covered, stories that deal with the topic are represented in Cyc; we then pose questions that any reader ought to be able to answer after having read the story.

The virtual secretary (ViSe) – an intelligent agent

The Virtual Secretary project is an on-going project at the University of Tromsø. It includes two phases: the first phase (ViSe1) focuses on user model-based software agents for information filtering and agent control propagation; the second phase (ViSe2) concentrates on information integration via co-operative agents in a distributed environment.

The Virtual Secretary is a software agent, which propagates in a network by the use of user models. The objective of the Virtual Secretary project is to construct software agents for secretarial tasks and that protect the users from unauthorized use of software agents (i.e., worm-like programs), and secure the propagation of such agents in a global network,. The first version of the Virtual Secretary will be capable of handling simple secretarial tasks as file retrieval from remote hosts and forwarding and filtering electronic mail (correspondence).

ViSe1 – User Model Based Software Agents

The user model is the core part of an adaptive user interface. A user model contains information of the user's past and current tasks, and enables personification of the user interface and application software. In order to respond to new actions, an adaptive user interface needs to detect what kind of action the user is executing, and how these respond to previous user actions.

ViSe2 - Information Integration via Co-Operative Agents in a Distributed Environment

In the ViSe2 project focuses on the construction of a multi-agent coopertaion system. As a research vehicle, we have chosen to build intelligent agents that perfor secretarial tasks for their users either by themselves or via cooperation. An individual ViSe2 agent has limited knowledge and problem-solving capabilities. Therefore, the agent has to interact with other agents to solve complex problems. In this sense, the agent's ability to reason about the other agents' activities and thus find the peer becomes a key issues.

Conclusion

There is a lot more work going on in this area, but due to lack of time only a few of them were listed. In short, Psychology can be of great help to making machines intelligent. There is a lot of potential for psychology in this area.

Thus, Psychology could help us understand common-sense reasoning and how knowledge is used in realistic tasks. How we deal with error and exceptions, and how we recover from them? How do we plan, and more directly how does the goal directed behavior work? How do we reason about time? And many more such questions.

To be honest, how much work Psychology has to do to answer these questions in a way that these are useful for Artificial Intelligence is a question that is yet to be answered.

References

1. ***The Emotion Machine*** by *Marvin Minsky* (Available at: http://web.media.mit.edu/~minsky)
2. **How Can Psychology Help Artificial Intelligence** by *Alvaro del Val* (Available at: http://www.ii.uam.es/delval)
3. *John Weng*'s Home Page (http://www.cse.msu.edu/~weng)
4. *SAIL* and *Dav* Home (http://www.cse.msu.edu/~weng/research/LM.html)
5. **A Theory for Mentally Developing Robots** by *J. Weng* (Published in: *Proceedings of the 2nd International Conference on Development and Learning)*
6. **The Developmental Approach to Intelligent Robots** by *J. Weng* (Published in: *Proceedings of the AAAI Spring Symposium Series, Integrating Robotic Research: Taking The Next Leap*)
7. **A Developing Sensory Mapping for Robots** by *N. Zhang, J. Weng* and *Z. Zhang* (Published in: *Proceedings of the 2nd International Conference on Development and Learning)*
8. Symposium on How to Design a Functioning Mind (http://www.cs.bham.ac.uk/~axs/aisb2000/)
9. *Cyc* (http://www.cyc.com/)
10. *OpenCyc* (http://www.opencyc.org/)
11. **Cyc: A Midterm Report** by *R. V. Guha* and *Douglas B. Lenat* (Published in: *AI Magazine* – Fall 1990)
12. **Cyc: A Large-Scale Investment in Knowledge Infrastructure** by *Douglas B. Lenat* (Published in: *Communications of the ACM* 38, no. 11)
13. *The Virtual Secretary* Home (http://www.vise.cs.uit.no/vise/)
14. **Using User Models in Software Agents: The Virtual Secretary** by *G. Bellika, G. Hartvigsen & R.A.Widding* (Published in: *Proceedings of the 3rd International Conference on Multi Agent Systems*)
15. **ViSe2 – An Agent-Based Export Consulting System with Efficient Cooperation** by *C. G. Bian, W. Cao & G. Hartvigsen* (Published in: *International Journal of Advanced Computational Intelligence*)