Superception

Complex, Low-Level and High-Bandwidth Perception in AGI-Aspiring Reasoning Systems

Jordi Bieger, Icelandic Institute for Intelligent Machines, jordi@iiim.is

Artificial General Intelligence is the study of the development of systems that can achieve their goals with insufficient knowledge and resources, and independent of domain. It should be clear that every such system needs some way to sense the current state and some way to act on it to move it closer to its goals. What those sensors and actuators must look like depends on the domain in which the system will be used, and therefore falls outside the scope of the AGI system. Yet, although specific perception modalities shouldn't necessarily be included in a general architecture, a general architecture *should* be able to support any sensors that are deemed necessary.

In many real-world situations, an intelligent system might like to make use of visual data (other data types are conceivable, but I will focus in vision here). The sensors that would generally be used to deal with such information (i.e. cameras) tend to produce very low-level, highly complex perceptual data at very high throughput rates.

In my presentation I won't pretend to have the solution for how to deal with this, but I hope to spur the debate by talking about a number of different strategies that could be used:

1. **Using raw inputs as the system's sensors**
   Some AGI-aspiring architectures were specifically made for this and some even believe that intelligence will naturally emerge from advanced perception and action. Subsymbolic systems like DeSTIN and HTM seem especially suited to this approach. On the other hand, some proto-AGI reasoning systems - including AERA, NARS and parts of OpenCog – tend to work best with higher-level concepts. These systems are specialized to work with local (rather than distributed) representations, and like all other systems are limited in resources (and could be overloaded by millions of pixels per second). The complex, low-level data will make representation hierarchies complex and deep, which can lead to several problems (such as training data requirements).
2. **Using an external module to extract higher-level features**
   A specialized external system could extract higher-level features like lines, curves, shapes, objects or even words (in written text). These could then be sent to the AGI reasoning system which would significantly decrease the complexity, low-levelness and throughput (especially since this external system could throttle it). These systems could come about in two ways: A) they could be hand-coded by humans who impart it with their knowledge of computer vision or B) it could be another learning system that is more suitable for vision than the architecture. An interesting special case of (B) is to use another instance of the given AGI-system. This would have some of the drawbacks of Option 1, but they would be somewhat diminished since both systems can specialize on their own tasks.
   One problem with this approach is that it will very likely remove data that might be useful to the system. A picture isn't just a collection of very clear lines. There is much more to it. We can also see this in other domains: sound is much more than pitch and amplitude, and language is much more than a bag of words.

This kind of feature extraction approach has been used in AERA, NARS and probably many other systems. It could be argued that OpenCog utilizes DeSTIN in this roughly way, although OpenCog fully integrated DeSTIN (Option 6; see also Option 5).

3. **Outsourcing perception to another module**
   This can be considered a special case of Option 2 (if you use really high-level features), but deserves to be mentioned on its own. The problem with this approach is that most of perception isn't solved. We don't know how to do computer vision, sound interpretation or natural language processing. Many would even argue that these are AI-hard problems. Even if your AGI reasoning system works perfectly, it would still be handicapped by subpar perception. It seems that one of the major elements that's lacking from current computer perception systems is top-down feedback from an intelligent system.
   AERA used such an approach for speech to text conversion, and it is in general a useful approach to more quickly test out an AGI-aspiring system.
4. **Using a mixture of external modules**
   By using a mixture of external modules, some of Option 3 and 4's problems might be mitigated at the cost of extra computation, and under the condition that the external modules' mistakes occur under somewhat differing circumstances. In this case, a generally intelligent reasoning system could discover the patterns in the modules' suggestions and might be able to infer the correct percept.
   To my knowledge, nobody is doing this.
5. **Interact with a more specialized module to "perceive together"**
   The difference between this approach and Option 3 and 4 is that the interaction between systems goes both ways. That way, top-down feedback from the intelligent reasoning system can aid the perception system, which I believe is the main missing component in computer perception. To facilitate this, it would be useful if both systems spoke the same "language" even if their internal workings would be more specialized to their task (i.e. perception vs. reasoning).
   OpenCog has done something like this by integrating DeSTIN.
6. **Integrate the ability for perception into the architecture**
   Some cognitive architectures have special internal modules for different kinds of perception. If the architecture contains special modules for e.g. vision and natural language processing that aren't general to the types of data that they process (i.e. complex, low-level, etc.), this could be considered "cheating" with respect to our definition of domain-independent intelligence (although probably not *their* definition).
   This approach is taken by most systems that aim to emulate humans in some capacity. OpenCog took this approach by integrating DeSTIN, although it could also be considered an example of Option 5. This kind of perception could also be implemented in AERA as part of the Masterplan, although the feasibility of implementing extremely complex systems (such as computer vision) here could be questioned and some of the objections in Option 1 still apply.

At this time, I believe Option 5 holds the most promise for systems such as NARS and AERA, since it would allow them to preserve – or perhaps even extend – their generality and domain-independence. Some of what has been said here could also be extended from perception to action, since action is said to be inverse perception and one can imagine similar issues with controlling a highly complex robot body in real time by sending electricity to the motors.

This talk is meant to elicitate discussion and not to present any solutions, so I hope everybody will contribute their ideas.