Innovation Paths for Machine Learning in Robotics

By Freek Stulp, Michael Spranger, Kim Listmann, and Stéphane Doncieux
Based on interviews with Moritz Tenorth, George Konidaris, Pieter Abbeel

Advances in artificial intelligence (AI), especially in machine learning (ML), are changing the business models of many companies, and creating entirely new ones. Recent research estimates that AI could boost profitability rates by 38% worldwide, leading to an economic boost of €12 trillion across a variety of industries by 2035. This immense number is an accumulation of many smaller numbers, related to the successful deployment of ML at individual companies, including small- and medium-sized enterprises (SMEs) and start-ups.

On the other hand, it appears that 40% of Europe's self-proclaimed AI start-ups do not use AI technology in a way that is essential to their business. In a recent IEEE International Conference on Intelligent Robots and Systems (IROS) workshop and the ongoing European project "VeriDream, (VERtical Innovation in the Domain of Robotics Enabled by AI Methods)" we address several questions related to the successful deployment of AI methods in robotics companies, especially concerning ML methods that have resulted from academic research projects. These questions include the following:

- How can start-ups manage the transition from novelty-driven research to identifying and satisfying customer needs?
- What are the consequences of building on academic open source software, and what is the impact of the software license chosen?
- How can the long tail of failures that impact safety and reliability, particularly important in robotics, be effectively addressed?

In this "Industry Activities" column, we summarize the main insights and personal experiences of three leaders in the field of robotics who have earned a name for themselves both as academic researchers and as successful innovators, based on a set of interviews at the aforementioned IROS workshop. We believe these insights to be valuable to companies who are contemplating the deployment of ML, as well as researchers in academia whose aim it is to one day found a start-up. This article categorizes the use cases and challenges related to ML methods, describes recommendations for spinning off a start-up on ML and robotics, and highlights that reliability is often the main bottleneck for successful innovation. (See "Terminology.")

Which Application Use Cases are Appropriate for ML Methods?

In the domain of AI, the concept that speaks to the imagination the most is "artificial general intelligence," i.e., a system that is able to achieve human-level intelligence or beyond. We believe such systems will remain science fiction for quite a while to come. As Moritz Tenorth mentions: "I'm skeptical that you can really have one system that learns it all. And the other question is, even if you could do this, should you do this?" (See "Moritz Tenorth.")

In practice, the first choice for achieving reliable intelligent behavior is still engineering as "powerful engineering enables reuse, easy configuration, and easy debugging during operation," according to Tenorth. On the other hand, problems for which no engineering solution is known are candidates for ML methods. George Konidaris describes the thought process as follows: "This is a hard problem and I don't know how to solve it. Therefore, I will use learning because learning is more efficient than trying to engineer

Terminology

We consider AI to be a property of a system, i.e., whether a system is able to display intelligent behavior or not. There is no common agreement on the exact definition or scope of the term, but it usually includes the ability to perceive, plan, reason, learn, and interact with humans. Common approaches for achieving intelligent behavior in systems include automated reasoning, optimization, evolutionary algorithms, statistical learning methods, artificial neural networks, and deep learning. The latter three belong to an important subclass of AI methods called *Machine Learning*, where tasks and representations are learned from data. The main focus of this article is on ML.

Digital Object Identifier 10.1109/MRA.2022.3213205
Date of current version: 2 December 2022

it. This is what we see in computer vision." (See "George Konidaris.") Tenorth confirms: "We do use ML in several cases, mainly for perception, and typically for isolated problems." This approach—what is known need not be learned, and what is not known must be learned—is summarized nicely by Tenorth: "We often know how the robot should behave because it's determined by law, by processes, or by some other specification. Then you have the choice of either programming it yourself or collecting data and reverse engineering the learning process so that the outcome of the learning process is what you had in mind beforehand. And that is the learning problem upside down."

The fact that robots act and make decisions makes them such a challenging application domain. As Konidaris explains, "Robotics, because of its sequential decision-making aspect, is fundamentally harder than supervised learning." So indeed, we see that much of the ML successes in robotics have been achieved in problems that can be formulated as supervised learning problems, such as computer vision.

If ML is the method of choice to solve an application task, workflows for data acquisition are essential. As Pieter Abbeel puts it, "A big part of our job is collecting a lot of data." (See "Pieter Abbeel.") Simulation and simulation-to-real approaches are important tools to facilitate data collection. Care should be taken to focus on the real problem at hand, rather than the simulated problem. The way Konidaris sees it, "What you find when you build the simulation is that you end up cobuilding the simulation, the benchmark, and the learning algorithms so that everything sort of works out together. Real robot hardware keeps you honest."

To summarize these insights and comments, innovation with ML methods in robotics will currently most likely be successful for isolated problems that can be formulated as supervised learning, and in which data acquisition is not too expensive, or can be automated with simulators that mirror the robot hardware sufficiently well.

Innovation: From Research Ideas to Products and Services

Innovation is the process of turning ideas into products or services. The transition from academic researcher to innovator thus requires a shift of perspective. As Tenorth explains, "Before I looked at it from a research point of view. What is novel? What hasn't been researched beforehand? Also, what can you write a paper about? In a company, it's about what works. Maybe a simple heuristic is good enough in some cases and saves you a lot of complexity." All the interviewees corroborate the following: "From day one, make sure that you work hard on understanding your customers. It is critical to know what really matters to them," explains Abbeel. From Konidaris' perspective, "If you're not talking to customers, then your symbols are all ungrounded when you're thinking about application." Understanding customer needs requires a substantial effort that does not involve technological progress

Moritz Tenorth

Moritz Tenorth is the CTO at **Magazino GmbH** (Munich, Germany). He received his Ph.D. from the Technische Universität München in 2011 on the topic of "Knowledge Processing for Autonomous Robots."

"I was very impressed by how much is possible if everyone really has a common goal."

Magazino GmbH develops and builds intelligent, mobile robots for intralogistics. Their products include the TORU robot for autonomous shoe box picking, and SOTO, for industrial production supply. AI methods are used in the overall architecture called *Advanced Cooperative Robot Operating System*. ML is used for individual problems, especially perception.

George Konidaris

George Konidaris is director of the Intelligent Robot Lab and an associate professor at Brown University (Providence, Rhode Island, USA). He is the chief roboticist at **Realtime Robotics**. His Ph.D. thesis was "Autonomous Robot Skill Acquisition" at the University of Massachusetts Amherst in 2011.

"Real robot hardware keeps you honest."

Realtime Robotics offers products for the intuitive programming of automated, collision-free motion plans and real-time control to execute these plans as well as spatial perception capabilities for dynamic unstructured environments. AI methods are used for motion planning and perception.

Pieter Abbeel

Pieter Abbeel is director of the Berkeley Robot Learning Lab and a professor at the University of California, Berkeley. He is a cofounder of Gradescope and **covariant.ai**. He received his Ph.D. from Stanford University in 2008 with the thesis "Apprenticeship Learning and Reinforcement Learning With Application to Robotic Control."

"From day one, make sure that you work hard on understanding your customers. It is critical to know what really matters to them."

Robots at covariant.ai learn general abilities such as robust 3D perception, physical affordances of objects, few-shot learning, and real-time motion planning. This allows them to adapt to new tasks just like people do: by breaking down complex tasks into simple steps and applying general skills to complete them.

on the robotic system. According to Abbeel, "For Covariant, we actually spent the first half of the year meeting with 200 other companies and asking them, 'If they could get a smart robot in their facility, what would that smart robot be doing?'"

Can researchers in ML and robotics prepare, already during their academic careers, for a potential future transition from novelty to customer needs? First of all, gathering experience in teams working on integrated systems is certainly beneficial. As Tenorth explains, "One important thing I learned also during my Ph.D. time is how to build a system, rather than only one part of a system. Other groups focused on individual algorithms, but we did things such as making pancakes with a mobile manipulator. In these cases, it was important that components are put together in a way that it results into an integrated system. I think that's the most important learning experience."

Second, open source publishing of code and the choice of license should be done with foresight. More companies continue to use and contribute to open source, i.e., according to Tenorth, "Magazino also releases open source software to the outside. So we try not to just consume but also to give back to the community." The choice of license has a big impact on potential innovation paths. Tenorth goes on to say, "The Apache, BSD (Berkeley Source Distribution), and MIT (Massachusetts Institute of Technology) licenses are good. Copyleft licenses such as General Public License are problematic because if you link against them in your system, the copyleft clauses apply to all other code that depends upon it. This implies having to share your source code with your customers, which is something that investors are very sensitive about."

Finally, when spinning off the start-up, "it's detrimental if you come up with a solution and try to find a problem that fits. It's better to see which problems need to be solved and which solutions are appropriate," says Tenorth. In this context, Konidaris recommends to "make sure to include some hard-nosed industry people in your spin-off."

Reliability Is Key

What sets robots apart from other digital agents is that they physically interact with the real world. This radically changes the requirements on reliability and safety, and these requirements are not always compatible with trial-and-error ML and extrapolation to novel situations. "Learning is very tricky for an industrial robot that literally weighs a ton and is working on a production line where the costs of downtime are in the tens or even hundreds of thousands of dollars a minute," says Konidaris. In this context, Abbeel corroborates the previous remarks by Tenorth on the divergence of academic requirements (novelty) and industrial requirements (reliability): "At Berkeley, we try to do things that have never been done before. If we can do it for the first time, that's great. It doesn't matter if it only succeeds 50–60% of the time. It's already a big deal because it was never done before.'

Which strategies are used to ensure reliability in robotic systems that use AI methods, especially data-driven ML?

The first strategy has already been mentioned, namely, that safety, reliability, and efficiency requirements are facilitated by considering isolated problems.

A second strategy is to have safety layers and fallback solutions around the ML algorithms. As Tenorth says, “Our vertically integrated system gives us some freedom in building fallback solutions. For example, we have a dedicated safety controller that allows us to navigate safely next to humans without having the whole software stack safety certified.” Implementing fallback solutions is essential, but they have diminishing returns. According to Abbeel, “If you need to rely on those fallback options too often, then it becomes too costly. So you need to have high reliability such that your fallback is triggered only very sporadically. And that way you create real value.”

These diminishing returns lead to a long tail of errors and failures that need to be addressed, which is costly and time consuming. As Tenorth shares: “I was very impressed by how much is possible if everyone really has a common goal. For instance, I was really surprised how fast we had some pick-and-place demos running after a few months. But what you really underestimate is how long the time is to get it reliable.” “As Moritz, I was initially surprised by how much we could get done with a relatively small amount of money. But then as we grew and engaged with more customers, I was also surprised by how much extra engineering we had to do to keep them all happy,” says Konidaris.

The final solution is to consider an entirely new approach entirely for a robotic system or one of its components. Should the system be evolved to become more reliable through engineering, or rather, should time and money be invested in trying out revolutionary ML methods?

As an example, the success of simultaneous localization and mapping (SLAM) hinged on the pervasive use of probabilistic representations, revolutionary at the time. On the other hand, “When SLAM was considered ‘solved’ in academia, it was probably 5–10 years before you could even think about deploying it in industry because it needed to be much more reliable and work off the shelf,” explains Konidaris. Whether to revolutionize or evolve to create value is one of the most challenging decisions to make, in ML and beyond.

Conclusion

From the interviews presented in this article, we learned that some of the key obstacles for deploying ML methods in robotics are the following: the unpredictability of the engineering effort required to address the long tail of potential errors; the uncertainty about whether, after all this effort has been made, the end result actually creates value for customers, or still relies heavily on suboptimal fallback solutions; and the fact that these two questions require both deep technical expertise and a “hard-nosed industry” perspective.

Academic robotics researchers considering a spin-off using ML methods ideally master not only the technical aspects of their field but also have the perseverance for dealing with long error tails and the right intuition about whether to invest time in engineering or a different (ML) approach altogether. They also should have the willingness to invest substantial effort on the nontechnical task of understanding which product creates value for prospective customers.

The interviewees have demonstrated that they are able to perform this balancing act between research and industry, and we are grateful to them for sharing their experiences and insights.

Acknowledgment

This work was partially funded by the European Union’s Research and Innovation Program under grants 951992 (VeriDream) and 101070596 (euROBIN). We thank Andrea Gigli for the transcription of the interviews.