Generalization in Neural Network Algorithms:
With a Focus on Robotic Applications

Presented by Priyanka Krishnamurthi

April 11th, 2018

Readings

1) Troy D. Kelley and Lyle N. Long, "Deep Blue Cannot Play Checkers: The Need for Generalized Intelligence for Mobile Robots," Journal of Robotics, vol. 2010, Article ID 523757, 8 pages, 2010. doi:10.1155/2010/523757
2) C. Devin, A. Gupta, T. Darrell, P. Abbeel and S. Levine, "Learning modular neural network policies for multi-task and multi-robot transfer," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017, pp. 2169-2176. doi: 10.1109/ICRA.2017.7989250
3) Erhan, Dumitru et al., "Scalable Object Detection using Deep Neural Networks" arXiv:1312.2249 (2013)
4) Lenz, Ian et al., "Deep learning for detecting robotic grasps" http://journals.sagepub.com/doi/pdf/10.1177/0278364914549607 (2015)

Abstract

This presentation takes an application-based approach to the topic of generalization in neural networks, looking particularly at the field of robotics. It begins with a general overview of the topic of generalized intelligence, using Deep Blue as an example of a supercomputer that could only perform a very specific task. From here, different techniques for generalization are explored—including but not limited to a redefinition of symbolic and subsymbolic architectures, subsumptive architectures, and cognitive architectures. The next three papers are focused heavily on particular applications of this idea in the past five years, each attempting to solve a robotic task while providing a certain degree of generalizability in their algorithms. These approaches include a modified reinforcement learning algorithm that relies less heavily on data collection through the use of transfer learning (Devin et al. 2017), a scalable approach to object detection using deep neural networks (Erhan et al. 2013), and a two-tiered deep network system used to detect robotic grasps (Lenz et al. 2015).

Spotlight Questions

1) What other applications (within or outside of robotics) require algorithms that are generalizable?
2) It is often difficult to strike a balance between having a specialized algorithm that performs very well on a specific dataset and having a more general algorithm with better performance across multiple tasks. Where does the perfect middle ground lie, in your opinion? Are there any

examples of algorithms that do this well? Alternatively, is it possible to achieve both?