Predictive Collaborative Manufacturing Robotics via Deep Reinforcement Learning

Scott G. Barnes and Adam M. Wickenheiser
Department of Mechanical & Aerospace Engineering, The George Washington University

Objective

The objective of this research is to develop, train, and implement a deep reinforcement learning based collaborative robotic system capable of learning to perform manufacturing tasks in collaboration with people

Introduction

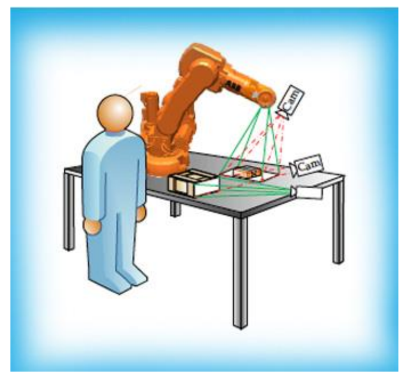

Figure 1. Collaborative Robotic Workspace

Despite recent advances in robotics, fewer than ten percent of production tasks are automated [1]. The complexity of most manufacturing tasks has made them unfeasible for traditional automation, resulting in the need for skilled human labor. In many situations, however, a blend of human and autonomous capabilities may prove to be safest, most efficient, and most reliable solution.

Interest in this middle ground has spawned the field of *collaborative robotics*, which seeks to enable people and robots to work together in a shared workspace. Collaborative robotics combines the repeatability and strength of robots with the creativity, dexterity, and problem-solving skills of people to accomplish complex tasks more efficiently [2]. While the established capabilities of collaborative robots allow people and robots to work in the same space, they do not enable collaboration in a manner comparable to that which occurs between people. Anti-collision and other safety features are important aspects of collaborative robotics, but simply avoiding accidents hardly qualifies as collaboration. Enabling true collaboration between people and robots requires significant expansion of the current capabilities of these robots [3].

This research seeks to advance the current capabilities of collaborative robots by establishing a machine learning framework to teach deep neural networks to learn to guide a robot to perform manufacturing tasks in collaboration with people. This approach will enable collaborative robots to become predictive rather than reactive, improving safety and increasing efficiency.

Background & Motivation

The fundamental concept behind the push for collaborative robots in manufacturing is the ability to maximize overall efficiency through division of labor by skillset. Robots excel at performing tasks which are highly repeatable, require high levels of precision, or put people at risk. Humans, on the other hand, perform best at tasks which require the ability to perform analytical problem solving or adapt to new situations. The ideal manufacturing environment is one in which these capabilities are utilized together to optimize overall efficiency and reliability while maintaining safety.

Developing mathematical models for human-robot interaction would be an impractical, costly, and non-scalable endeavor. Instead, we propose a solution in which models and policies are learned by neural networks, a parallel computing structure modeled after the human brain, in which a system of interconnected neurons are trained to learn a complex function from data.

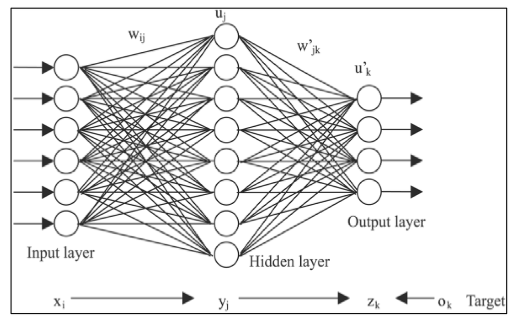

Figure 2. Artificial Neural Network

Computer Vision Software

Computer vision software was developed in the MATLAB Image Processing Toolbox to extract important features from video frames taken from the workspace. In applications such as collaborative robotics, in which images must be processed at a high enough frame rate to detect changes in people's movement rapidly enough for the system to react in a timely manner. Therefore, image processing was done in two phases. The initial frame was analyzed according to the process depicted in Figure 3.

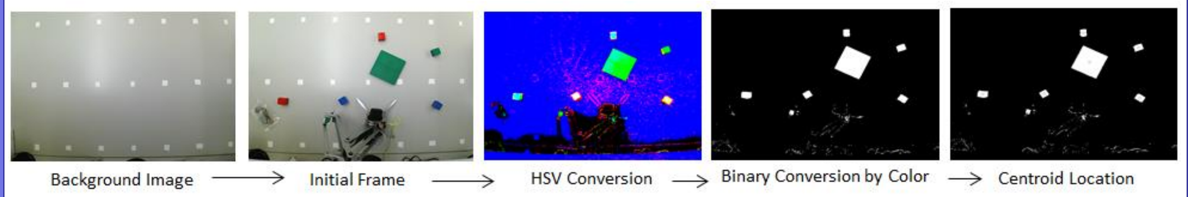

Figure 3. Initial Image Processing Algorithm

For each subsequent frame, background subtraction was performed once on the HSV converted image, followed by connected pixel calculations, resulting in the location and size of each object. This technique allowed for real time image processing at a rate of 20 frames per second. This high frame rate is critical to ensuring the system's ability to quickly react to changes in the person's behavior.

Deep Q Learning Algorithm

This research was inspired by DeepMind's research in applying Deep Q Learning to learning Atari 2600[4]. We modify their learning algorithm to utilize memory replay as batch training, enabling improved convergence to a proper decision policy (Eq. 1).

1. Initialize action value function Q with random weights and biases
2. Observe initial state S

Repeat:

  Initialize Replay Memory M

  Repeat:

  1. Selection action A, with probability
     ε that A is random
     1-ε that A = $\underset{A}{\operatorname{argmax}}[Q,(S',A')]$
  2. Carry out action A, observe new state S' and reward R
  3. Store experience (S,A,S',R) in M
  4. Update Q Network with (S,A,S',R) by minimizing the loss function

$$L = \left(R + \Upsilon \max_{A}[Q(S',A')] - Q(S,A)\right)^2$$

  5. $S \leftarrow S'$

  Until Simulation Termination

  Update Q Network on M using Lavenberg-Marqeardt backpropagation

Until Convergence

Figure 3. Initial Image Processing Algorithm

This algorithm was applied to train Deep Q Networks in simulations of manufacturing-related tasks.

Human Behavior Prediction

An important facet of collaborative robotics that has yet to be addressed is the ability learn people's behaviors when performing a specific task. Training a neural network to predict human behaviors can be used assist reinforcement learning algorithms in learning more robust decision policies that result in safer actions.

A pattern recognition feed forward neural network was build in the MATLAB Machine Learning Toolbox. This network's output is constrained by a Softmax layer (Eq. 1), which learns a probability distribution.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} j = 1, \ldots, k \qquad (1)$$

The network was trained to learn to predict a person's based on the first ¼ of the person's trajectory and the location of the two random locations the person could select to go to. In these conditions, the network was able to predict the person's goal location correctly 97.0% of the time.

To demonstrate the scalability of this approach, we then trained the network on data in which the person had eight possible locations to choose from, resulting in the network making accurate prediction 90.3% of the time.

Simulation: Training, Testing, & Results

In order to train the Deep Q Network to learn to perform new tasks in collaboration with a person, simulations were built in which the network acted through an agent to simulate the movements of the end effector of a robotic arm. The primary task the robot was trained to perform was a basic assembly task, in which six parts were added to a larger component.

The Deep Q Network was trained on this simulation, in which 100 unique workspaces were generated with pseudo-random initial conditions. The feed forward artificial neural network (Fig. 2) was trained according to the our modified Deep Q Learning algorithm (Eq.2) using Levenberg-Marqeardt backpropogation (Eq. 2), which finds the parameter β of the function f(x,β) that minimizes the mean squared error.

$$\beta = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{m} [y_i - f(x_i, \beta)]^2 \qquad (2)$$

Where m is the set of empirical datum pairs x_i and y_i

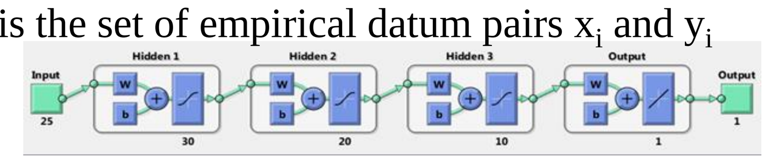

Figure 2. Deep Q Network Structure

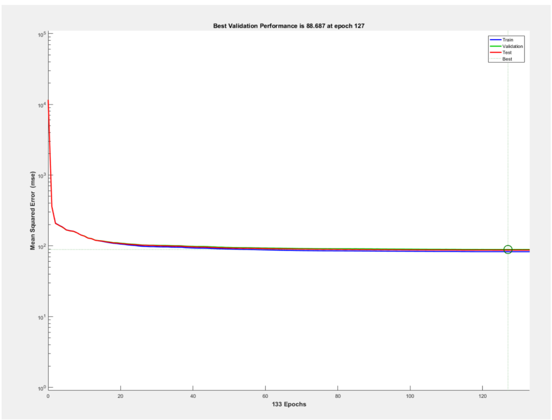

Figure 3. Mean Squared Error per Epoch

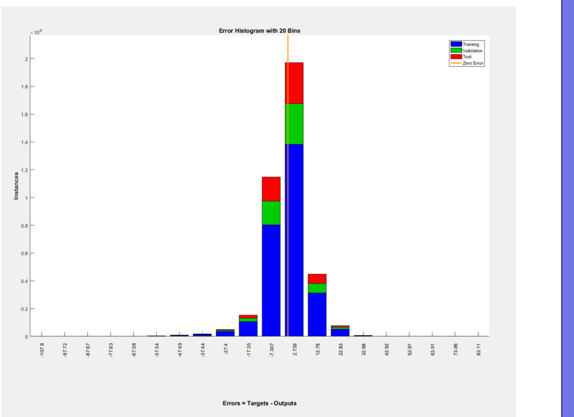

Figure 4. Network's Final Error Distribution

The network was then tested on fifty newly generated simulations with pseudo-random initial conditions. The network guided the simulated robot to increase the efficiency of the process by 28% over the person working along, while directing it to take safe actions 87.9% of the time.

Robotic System: Testing & Results

The computer vision software and trained Deep Q Network were then integrated with uArm Metal (Fig. 8), a desktop robotic arm modeled after a manufacturing robot. The system was then tested on its ability to perform the assembly task it had been trained to learn.

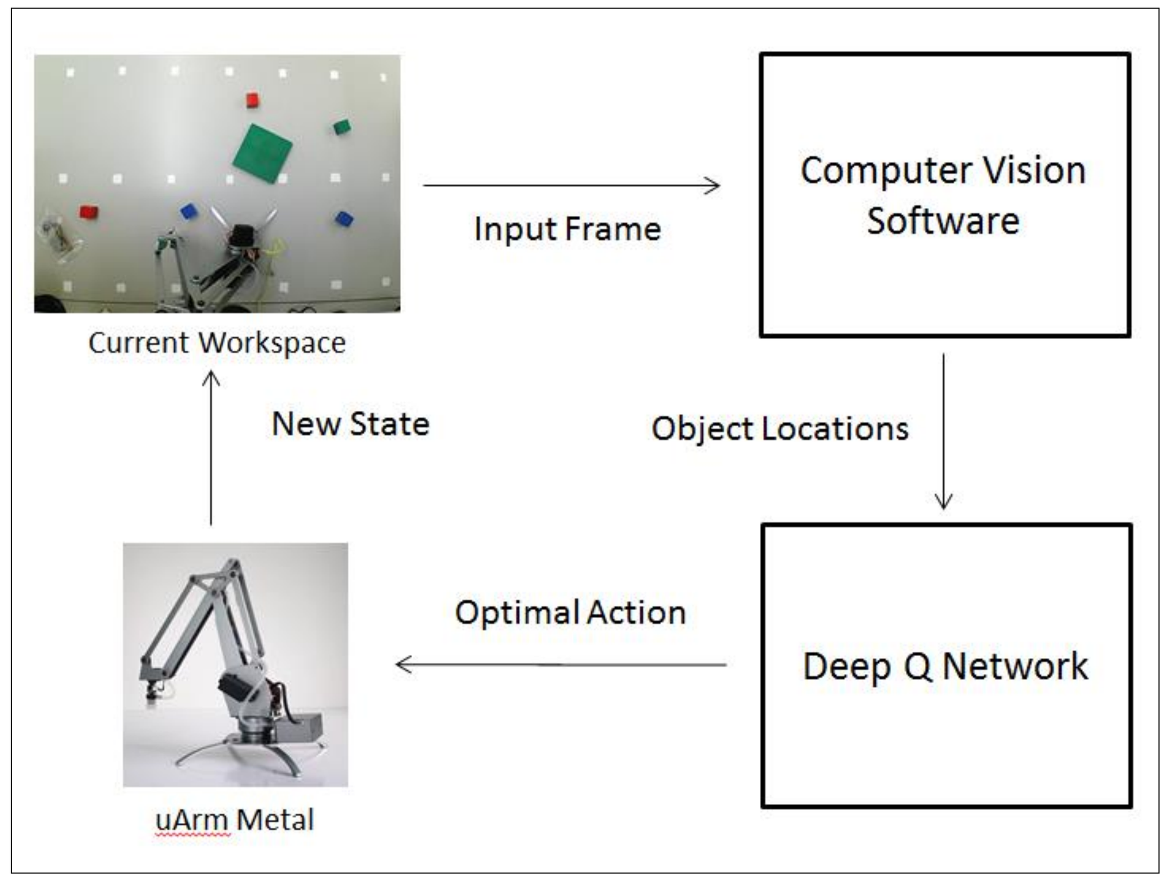

Figure 8. Adaptive Control System Architecture

Results from these test will be included in the final presentation.

Analysis & Discussion

One of the benefits of Deep Q Networks is that they learn optimal decision policies on their own, allowing for the possibility to solve problems in ways that people have not thought of. This became evident in some of the initial training runs, in which the network learned an optimal decision policy in conflict with our preconceived notion of how the task was to be completed. When planning this task, we envisioned that one part at a time would be brought to the central component for assembly. However, the robot learned to avoid collisions and maximize its efficiency by assembling multiple parts away from the main component, and then attach the assembled parts to the main component.

Conclusion

This research successfully demonstrated the feasibility of utilizing deep machine learning to direct a collaborative robot to work safely and effectively with people. This framework was successfully implemented to learn an assembly task, resulting in improved overall efficiency while maintaining a high standard of safety.

The networks' performance in a number of ways. Because Q Learning assumed a Markov Decision Process, a feed forward neural network is insufficient to fully define dynamic states, such as those that involve the motion of people and robots. This can be resolved by adding recurrence to the network, enabling the network to 'remember' previous information [5].

Future Work

Building on the results of this work, we propose a fully integrated system of neural networks to distribute the process of learning complex tasks across multiple networks. It will be capable of learning to identify and classify important objects in a manufacturing workspace, learning manufacturing processes and the specific tasks associated with those processes, learning human behaviors associated with those processes, and planning safe trajectories in order to safely and efficiently collaborate with people. This system will incorporate real time user feedback and a database of trained networks and learned behaviors to learn more robust policies and ensure the system's performance.

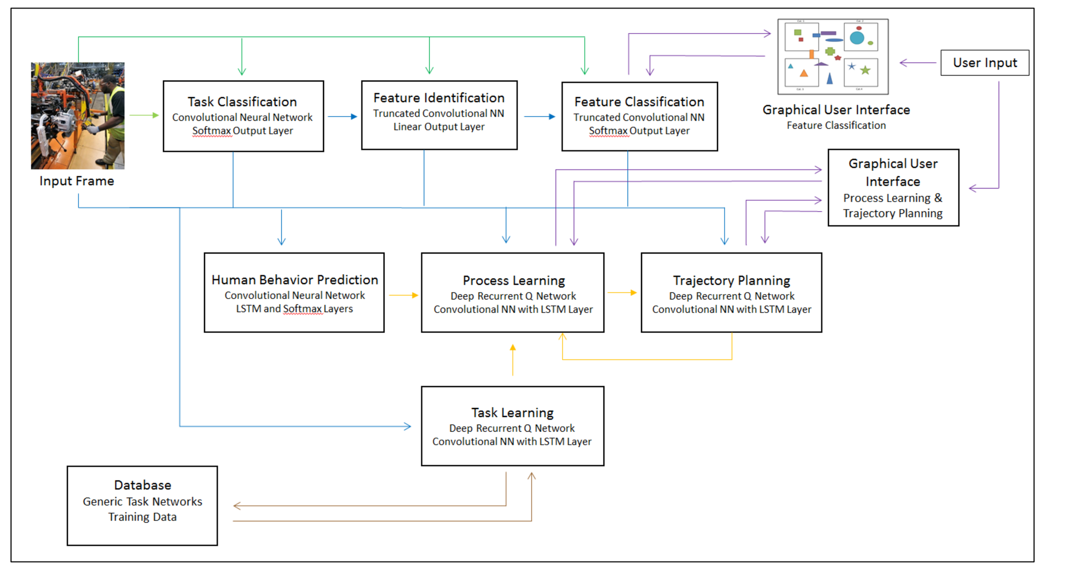

Figure 9. Proposed Deep Learning System Structure

References

1. M. Chui, J. Manyika, and M. Miremadi, "Where Machines Could Replace Humans - And Where They Can't (Yet)," *McKinsey Quarterly*, McKinsey & Company, July 2016.
2. J. T. C. Tan, F. Duan, Y. Zhang, K. Wantanabe, R. Kato, and T. Arai, "Human-Robot Collaboration in Cellular Manufacturing," *IEEE International Conference on Intelligent Robots and Systems*, 2009.
3. M. Bélanger-Barrette, Collaborative Robots, Risk Assessment: An Introduction, Robotiq, March 2016.
4. V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," arXiv:1312.5602 [cs.LG], 2013.
5. Hausknecht, Matthew, and Peter Stone. "Deep Recurrent Q-Learning for Partially Observable MDPs." Cornell University Library, 11 Jan. 2017. Web. 01 Feb. 2017.