CS229: Machine Learning Project Final Writeup

Force Feedback Learning Applied to Robot Object Handling

Cason Male

The purpose of my machine learning project was to apply learning algorithms to a robotic hand, in order to better hold and pick up various objects. My project was part of STAIR or the Stanford Artificial Intelligence Robot whose next goal is to clean off a cluttered desk filled with objects the robot has never seen before. My project for Machine Learning was to use the existing STAIR software and hardware to implement force feedback control in the fingers of the robot. More specifically, my project focused on the classification problem of detecting when the robot was actually touching an object and when it wasn't. I worked with Quoc Le who helped guide me through this specific project task.

At first I didn't believe that applying learning algorithms would be necessary to solve the task; that adding resistive force sensors to the fingertips of the robot would be enough to detect when the finger is touching an object and when it is not. The problem occurs from the elastic hysteresis property of the 'skin' of the fingertip. The skin is an outer covering on the finger that transfers external forces to the underlying sensors within the finger. The elastic nature of the skin causes hysteresis when a force is loaded and unloaded to the finger. Elastic hysteresis means that the deformation of the skin is not consistent when loading and unloading a force. The material requires less force to deform the same amount when unloading as compared to loading. This means there is a period of time when the load is removed, the material remains deformed and slowly comes back to its original shape. What this means for the project is that although a load is removed from the finger skin, the skin remains deformed for some time after and so triggers the touch sensors beneath it. Without learning, unloading causes the sensors to detect a 'touch' even though the load from the object had already been removed. The only way to solve this problem was by optimizing the mechanical design and using learning algorithms to train the finger to know what a real touch is.

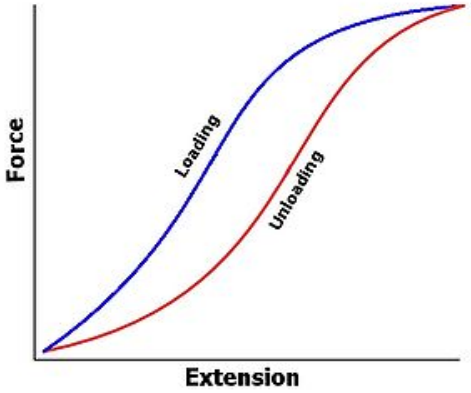

Figure 1: Elastic Hysteresis

The first half of my machine learning project was focused on perfecting the mechanical design. The fingertip design was a long process but proved to be time well spent to make learning process easier. Results from testing the force feedback in the sensors relies heavily on the design of the finger and finger skin which can be optimized to make it easier for the learning algorithm to find better solutions for touch and slippage when grasping objects. The model used in previous attempts to make a robot finger I slightly modified for the project but the basics are the same. It is important for the design to expose the sensors directly to the skin in a way that outside forces will be directed towards the underlying force sensors. In this way the surface that can detect contact should be maximized. It was

very important to keep the three sensors secured to the base of the finger directly without any elastic material between them and the finger model which was used before I began work on the finger.

Most of my work on the mechanical design was for skin fabrication for the fingers of the robotic hand. Many different materials were tested for the finger in order to find the right one for the application. I got to test and fabricate skins in the Mechanical Engineering lab with the help of Barrett Hyneman and John Ulmen from Mark Cutkosky's lab. The previously used material for earlier iterations of the finger skin was not strong enough to last many cycles of picking up small objects or even fewer cycles for heavier objects. The final material used is a TAP silicone RTV which is durable while providing a reasonable amount of force transmission to the underlying sensors. The biggest problem in fabricating the skins was actually getting a smooth cast without any air bubbles. Many different techniques were used in order to create a decent cast for testing. Modifications were also made to the fingertip mold itself to improve the skin shape. The previous model has a groove in the top for the skin to slip into as a way to provide a good fit on the finger without using any kind of adhesives. However, the grove in the finger and the notch in the skin were not compatible. To fix this, the groove made for the mold was removed so the skin is flat along the finger model everywhere except of course in the sensor region which has three pads made to contact with the force sensors.

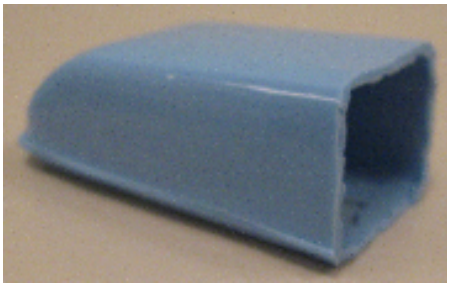

Figure 2: Finger Skin

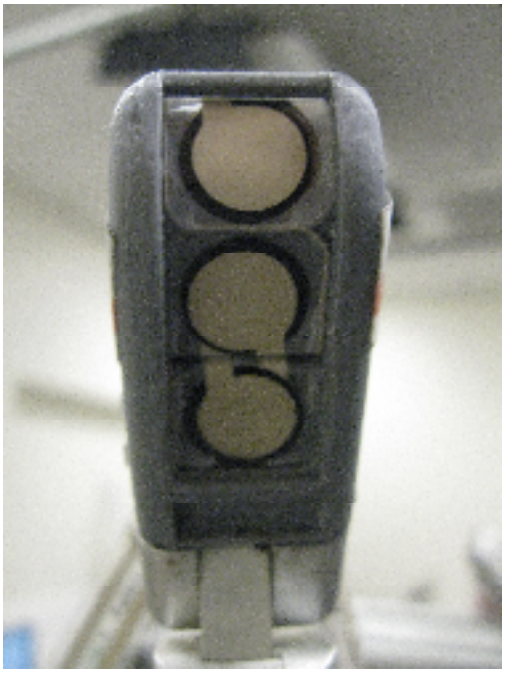

Figure 3: Finger and sensors

One of the other major problems with the original fingertip design was the attachment of the sensors. Some sort of adhesive tack was used previously which provides many potential errors for sensor feedback. Since the material was compliant, it acted as a cushion between each sensor and the finger model itself. This is not a good design since it brings a damping effect into the sensor reading, making it harder to collect real data that reflects the force acting on it. Instead, the tack was removed and each sensor was trimmed and super glued directly to the finger model. With a hard surface behind each sensor, force readings will be much more accurate since each sensor now takes on the direct force applied to the skin without compliance.

Once the mechanical design was robust enough and optimally designed for testing, sample data was needed for the learning algorithm to train on. Data collection was made by the Phidget USB input data device which was directly connected to each of the finger sensors. The Phidget has the capability of collecting data from up to eight sensors but for my project, only three were needed for the finger. To

classify the binomial distribution of the data, Support Vector Machines were the optimal choice using a linear kernel. Data was collected for the SVM by recording data sets that were pre-labeled to be a touch or no touch. To do this, I told the program that the data collected in the next X hours would have the 'touch' label which is 1. I would then press on the finger pad for a number of hours while data was collected. I repeated the process again for 'no touch' label -1 where I let the program record force sensor values in the finger for a number of hours without the finger having any external loads applied to it. This process took a number of iterations to collect adequate data for the SVM to learn properly. After the first iteration I tested the SVM which drew the boundary so that soft touches were not being detected by the finger. To fix this I recorded more 'touch' data, this time making sure I gently pressed the surface of the finger skin. After enough iterations of data collection the SVM was able to find a proper boundary to distinguish a real touch from a false one.

Integration of the SVM algorithm into the existing robot controller was the next task. The touch function I created in C initializes the phidget data hardware then reads in the touch sensor values which are then evaluated in the SVM algorithm. The values are then multiplied as an inner product with the generated SVM parameters: $\omega^T x$ with an intercept value of 0. If the resulting value is greater than zero than the function returns a touch. Otherwise if the resulting value is less than zero, a 'no touch' is returned. This function had to be written in C to use the phidget C libraries but the arm control program is written in Python. In order to get these programs to communicate, I used the Simplified Wrapper and Interface Generator, which simply wraps the touch function as a Python module to be used in the robot control code. The program is able to constantly loop and check for a touch event from the finger in real time.

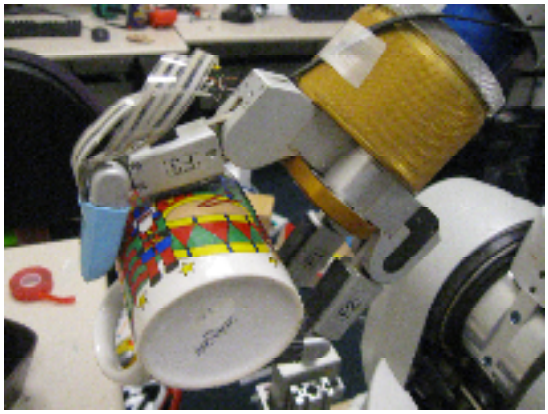

Figure 4: Hand gripping a mug with force feedback

Currently the control code monitors a single finger, since I was only able to fabricate one new finger for testing. The code does work well for the single finger and successfully reacts to the feedback in the finger tip when grasping objects. Before my work, the robot simply grasped its hand around an object until the motors could not clench the fingers any further. This has the potential for many problems, and from a mechanical standpoint, working the motors in this way is not good for the arm. After my feedback integration was implemented, the robot now stops closing its hand once a touch is detected by the finger. The other big fix due to the force feedback in the fingers is slip detection of an object from the robot's hand. Previously, if the robot grabbed an object and moved it to a new location, there was no way to detect if the object had slipped out of the hand during the transition. The finger sensors now prevent against this and similar problems since the robot can detect in real time if it is grasping an object or not.

The results of the first force feedback finger sensor were successful but there is a lot of potential for this new application and much room for improvement which I would like to pursue in the next quarters. One of these improvements would be to tweak the mechanical design of the finger and skin to

make a greater touch surface along the whole finger. Although the current version works, sensing is directed only along the top of finger and not along the sides. Side sensors will be critical for slip detection which is a huge area of potential for the finger sensing application. It is quite possible to get the robot to learn how to adjust for object slippage on the transition to prevent objects from falling. For instance, when sensors read a particular value, the robot should learn to reorient its hand or grab the object tighter. I also plan to build two more fingers to complete the hand. Once three fingers are in place, the controller will be adjusted to monitor each finger individually. In the future when the hand closes, each finger will stop when it detects its own touch and adjust as the object shifts within the grasp during movement. The last part will be to integrate this with the path planning and object recognition projects being currently worked on. I thoroughly enjoyed working on this project due to my passion for robotics and getting a chance to apply machine learning from class. Coming from Carnegie Mellon with a desire to continue my robotics studies at Stanford, it was a pleasure and privilege to be able to work with the famous STAIR robot and learn the fundamentals of machine learning in CS229.