6.857 Final Project: Milestone 6

Sebastiani Aguirre Navarro and Rachel Holladay

I. Introduction

RH: tweak introduction from before, expand upon it. add mention of "one shot"

II. Related Work

While we primarily build off of the Dex Net 2.0 paper [1], there is a wide range of literature investigating learning how to grasp objects. The overwhelming majority of the recent work has focus on using CNNs, although there are a few papers that use SVMs, kernel-density estimation and constrained optimization-based techniques [2], [3], [4].

Ten Pas et al [5] developed a similar grasp detection algorithm to DexNet paper by generating grasp hypotheses and training a 4-layer CNN to perform binary classification on whether the grasp is viable. They use a different grasp representation and rely on the BigBird data set [6]. Rather then classifying a grasp, Johns et al uses a CNN to learn a grasp function, which provides a score for each grasp. At execution time, then can compare the scores of several grasps and select the best grasp [7].

The above works focus on using a parallel jaw gripper, a two finger hand where the fingers are parallel to each other and usually move together. While this is a relatively simple hand, it is ubiquitous in industry and research and still allows for complex manipulation tasks. However, people have worked to expand this grasp prediction to more complex, multi-fingered hands using various CNN architectures [8], [9], [10].

Within the grasp learning community, and in fact, within the robotics learning community there is a pull between real data collected through a robotic platform and data generated from a physics simulator. While data collected on a robot better captures reality (since physics simulators are far from perfect), data collection is difficult and time-consuming. Pinto et al collected, at the time, a record amount of data at 50k data points of grasps collected across 700 robot hours [11]. Levine et al later collected 800,000 grasp attempts over a two month period, using between six and fourteen robot arms at once [12]. While these approaches allowed them to train a CNN without over fitting or simulation data, such data collection is not always practical and require a huge amount of engineering effort. Bousmalis showed how to augment a smaller amount of real data with simulation to improve accuracy, thus attempting to combine the merits of both [13].

While most of this work focuses on using color (RGB) or depth (RGBD) images [14], there is growing interest in using tactile feedback, inspired by how humans feel as they grasp. Calandra et al combines vision and touch sensing to build a visuo-tactile CNN that predicts grasp outcomes from a combination of the modalities [15]. This can go one step further in using tactile feedback to learn how to readjust while grasping [16].

Dex Net 2.0 is the seccond of three pieces of research. Dex Net 1.0 solves the same grasping problem, but uses a multi-armed bandit model to correlate the rewards of a proposed grasp with previously seen grasps. [17] The similarity metric between grasps is learned from a Multi-View CNN. using suction [18].

III. Data Set

We opted to use the Dex Net 2.0 data set due to its size, ease of use and parametrization [1].

Mahler et al define a generative graphical model defined over the camera pose, object shape and pose, friction coefficient, grasp, depth image and success metric. To generate the data set they make i.i.d (independent and identically distributed) samples from their generative graphical model, resulting in 6.7 million data points.

The data set is defined over 1,500 object meshes that were used in Dex-Net 1.0 [17], collected from a variety of other data bases and standardized with respect to position. For each object, they generated 100 parallel jaw grasps via rejection sampling of antipodal pairs and evaluated a robust epsilon quality grasp metric on each grasp [19]. Additionally, each object is paired with a rendered depth image (2.5D point cloud[1]) from the sampled camera pose.

The data set of 6.7 million data points has 21.1% positive examples. This is unsurprising, since it is much more difficult to find successful grasps, as compared to failed grasps.

From the data set we randomly sample, with replacement, k data points. In some cases we sample such that we guarantee some ratio of positive versus negative examples. Since we are sampling our data sets, we will sample multiple copies and average the final results.

IV. Problem Statement and Architectures

RH: Define input and output. Say using cnn. describe software. Describe one-shot learning. then say

[1]The images are 2D matrices that are referred to as 2.5D in robotics literature because they display depth information.

we investigated the following changes RH: Currently, the input format is a 32x32x1 depth map and a 1x7 pose vector.

V. Results

Use keras [20] Andreas Network [21]

For the following results, we use the balanced dataset with our input as the image for each data point and the 7-dimesional grasp vector. During training, 80% of the dataset was used as train set and the remaining 20% as test set. Below we describe and show the results of two achitectures, which we refer to as the Inception Network [22] and the ResNet. As discussed in Sec. VI, these are the some of the many networks we will be testing.

The inception network consists of 1 convolution layer in the beginning with 10 filters of size 3x3. The output of this layer is passed in parallel to three convolutional layers of sizes 1x1, 3x3, and 5x5, each with 16 filters. These outputs are concatenated on the depth dimension and passed through a max pooling layer of 3x3. The outputs are flattened with global average pooling and then the pose vector is concatenated before passed to a classifier of one hidden layer of 20 units, as seen in Fig. **??**.

The residual network consists of two convolutional layers, one with 8 filters of size 7x7 and the next with 16 filters of size 3x3. At this point, the output of this layer branches, such that this same output is passed through two more convolution layers of 32 filters 3x3 and 16 filters 1x1 used as dimension reduction. The output of these two layers is added to their input and then passed to another convolution layer of 8 filters of 1x1 for further dimension reduction and then flattened with global average pooling. Like for the other network, the pose vector is concatenated to this output before passing it to a classifier with a fully connected layer of 10 hidden units, as shown in Fig. **??**.

In Fig. **??**, we can see that for the inception netowrk, the training loss decreases while the validation loss, while oscilating, increases. The same trend is shown in Fig. **??** for the residual net. This means that, like the results in our previous milestone, the network is overfitting to the data. The final training accuracy for inception net and residual net are 70% and 75% respectively, while both do 50% on the validation set. One possible modification is to add regularization on the fully connected layers, or to modify the architectures by removing or adding more layers. We will continue to explore this as well as experimenting with the representation of the data.

References

[1] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[2] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from rgbd images: Learning using a new rectangle representation," in *ICRA*, pp. 3304–3311, IEEE, 2011.

[3] M. Kopicki, R. Detry, M. Adjigble, R. Stolkin, A. Leonardis, and J. L. Wyatt, "One-shot learning and generation of dexterous grasps for novel objects," *IJRR*, vol. 35, no. 8, pp. 959–976, 2016.

[4] IEEE, *Bridging the gap: One shot grasp synthesis approach*, 2012.

[5] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, p. 0278364917735594, 2017.

[6] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel, "Bigbird: A large-scale 3d database of object instances," in *ICRA*, pp. 509–516, IEEE, 2014.

[7] E. Johns, S. Leutenegger, and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," in *IROS*, pp. 4461–4468, IEEE, 2016.

[8] Q. Lu, K. Chenna, B. Sundaralingam, and T. Hermans, "Planning multi-fingered grasps as probabilistic inference in a learned deep network," in *ISRR*, 2017.

[9] J. Varley, J. Weisz, J. Weiss, and P. Allen, "Generating multi-fingered robotic grasps via deep learning," in *IROS*, pp. 4415–4420, IEEE, 2015.

[10] Y. Zhou and K. Hauser, "6dof grasp planning by optimizing a deep learning scoring function," in *R:SS Workshop*, 2017.

[11] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *ICRA*, pp. 3406–3413, IEEE, 2016.

[12] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *IJRR*, p. 0278364917710318, 2016.

[13] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, *et al.*, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," *arXiv preprint arXiv:1709.07857*, 2017.

[14] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *IJRR*, vol. 34, no. 4-5, pp. 705–724, 2015.

[15] R. Calandra, A. Owens, M. Upadhyaya, W. Yuan, J. Lin, E. H. Adelson, and S. Levine, "The feeling of success: Does touch sensing help predict grasp outcomes?," *arXiv preprint arXiv:1710.05512*, 2017.

[16] Y. Chebotar, K. Hausman, Z. Su, G. S. Sukhatme, and S. Schaal, "Self-supervised regrasping using spatio-temporal tactile features and reinforcement learning," in *IROS*, pp. 1960–1966, IEEE, 2016.

[17] J. Mahler, F. T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, T. Kröger, J. Kuffner, and K. Goldberg, "Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards," in *ICRA*, pp. 1957–1964, IEEE, 2016.

[18] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-net 3.0: Computing robust robot suction grasp targets in point clouds using a new analytic model and deep learning," *arXiv preprint arXiv:1709.06670*, 2017.

[19] D. Seita, F. T. Pokorny, J. Mahler, D. Kragic, M. Franklin, J. Canny, and K. Goldberg, "Large-scale supervised learning of the grasp robustness of surface patch pairs," in *SIMPAR*, pp. 216–223, IEEE, 2016.

[20] F. Chollet, "Keras (2015)," *URL http://keras. io*, 2017.

[21] U. Viereck, A. t. Pas, K. Saenko, and R. Platt, "Learning a visuomotor controller for real world robotic grasping using easily simulated depth images," *arXiv preprint arXiv:1706.04652*, 2017.

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, pp. 1–9, IEEE, 2015.