6.857 Final Project: Milestone 4

Sebastiani Aguirre Navarro and Rachel Holladay

November 9, 2017

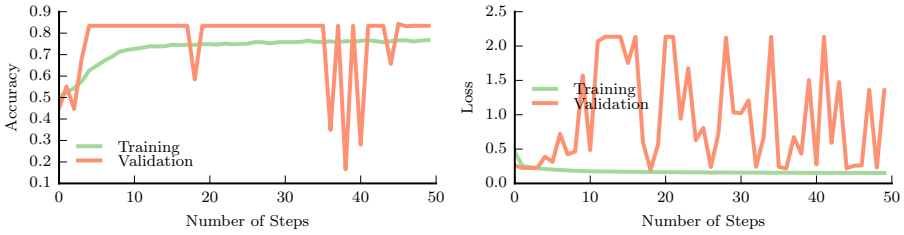
1 Initial Results

Our goal is to use machine learning techniques to predict the probability of successfully grasping an unknown object with a robotic arm. We were forced to change our data set, for reasons explained in depth in Sec. 2, to the Dexerity Network (DexNet) 2.0 data set [1]. We will first describe the input and output of our system and then describe some initial results. The DexNet 2.0 data set has 6.7 million synthetic point clouds with parallel-jaw grasps (a common robot hand type of two parallel fingers) and analytical grasp metrics.

The data set covers 1,5000 3D object models used in DexNet 1.0 [2] that are collected from a variety of other data bases and standardized with respect to position. Each object is paired with 2.5D point clouds, referred to as depth images, which are rendered with a variety of object and camera poses, where the camera intrinsics are known and used to center the depth image in a standardized fashion. Each image is a black and white 32 by 32 pixel matrix. The parallel jaw grasps were sampled with rejection sampling for antipodal point pairs. The grasps are represented by a 7 dimensional vector specifying details of the grasp center, angle, object center and gripper width. Together the depth image matrix and grasp vector compose our input.

Our label is given by the robust epsilon quality grasp metric (defined in [3]), which is thresholded by the value 0.002 to create binary labels.

For our initial results we sampled 10,000 data points from the 6.7 millon. Since the structure of the data changed considerably, pretrained models cannot be used since there is not an RGB component in the data. Therefore, we designed a simple Residual Network like the one shown in Fig. 2. The idea is to pass the 32x32 depth maps through a ResNet with Batch Normalization to extract 1x8 features from it, and then these features are concatenated to the hand pose features that are passed to a fully connected layer that ouputs the likelihoods of each binary class. The training was carried in Keras [4], using the RMSProp optimizer during 50 epochs with a batch size of 32 and 10% of the dataset was used as Test. Given that we had a class imbalance of 1858 positive vs 8150 negatives in our datset subset, we assigned weights of 0.8 for positive and 0.2 for negative such that the underrepresented class could influnce training during each gradient descent step. This initial architecture did not work



(a) Accuracy

(b) Loss

Figure 1: Initial Loss and Accuracy across the Training and Validation Sets

so well, however, since the error in the Test set was 81.0%. If we observe the curves for loss and accuracy on Train and Validation during training in Fig. 1b and Fig. 1a, we can see that the network is overfitting.

Moving forward, we have two ideas to address the overfitting issue. The first is to create a more balanced training set with respect to the positive and negative labels. Another option is to reduce the model complexity and train on a smaller subset. We will also continue to experiment with network architecture, representations and metrics.

2 Risk Mitigation

One of the risks we had mentioned in a previous milestone relating to data access unfortunately became realized. Early in our project we mentioned two possible datasets, the BigBird data set [5] and the DexNet data set [1]. We elected to use the BigBird data set in conjunction with the grasp generation and labeling process package "Grasp Pose Generator (GPG)" from [6]. In attempting to create our training data, we realized that GPG did not directly use the data in the BigBird set and instead first performed a transformation that editted the depth files and generated surface normals. We were unable to find the opensource component that performed this transformation and the author of the report (at the time of this writing) has not replied to our request.

Therefore, we have switched to using the DexNet 2.0 data set, whose data is described in further detail in Sec. 1. If we are able to use the BigBird data set, we will consider using it in conjunction or as a comparison.

3 Division of Labor

Rachel generated the training data from the DexNet data sets and set of the labeling mechanism. Sebastiani set up an environment in a AWS with GPUs, coded and trained initial Network In Network, Inception and ResNet based models in Keras/Tensorflow.

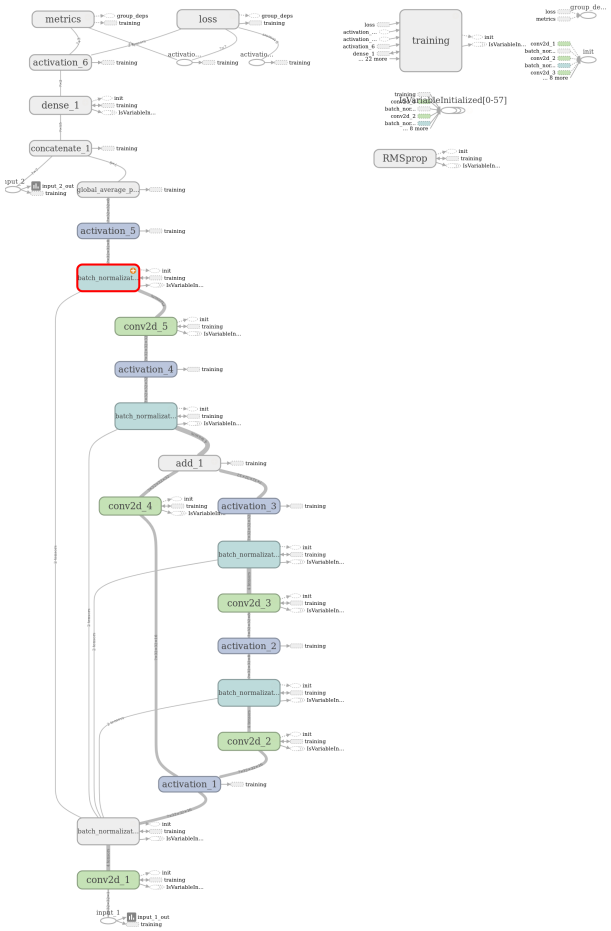


Figure 2: Network Structure

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] 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.

[3] 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.

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

[5] 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.

[6] A. t. Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *arXiv preprint arXiv:1706.09911*, 2017.