Cite as: Hadi Beik-Mohammadi, Matthias Kerzel, Michael Görner,
Mohammad Ali Zamani, Manfred Eppe, Stefan Wermter
IROS 2018 Workshop on Machine Learning in Robot Motion Planning - Sep 2018.

Neural End-to-End Learning of Reach for Grasp Ability with a 6-DoF Robot Arm

Hadi Beik-Mohammadi[1], Matthias Kerzel[1], Michael Görner[2], Mohammad Ali Zamani[1], Manfred Eppe[1] and Stefan Wermter[1]

***Abstract*— We present a neural end-to-end learning approach for a reach-for-grasp task on an industrial UR5 arm. Our approach combines the generation of suitable training samples by classical inverse kinematics (IK) solvers in a simulation environment in conjunction with real images taken from the grasping setup. Samples are generated in a safe and reliable way independent of real robotic hardware. The neural architecture is based on a pre-trained VGG16 network and trained on our collected images as input and motor joint values as output. The approach is evaluated by testing the performance on two test sets of different complexity levels. Based on our results, we outline challenges and solutions when combining classical and neural visuomotor approaches.**

I. INTRODUCTION

Reach-for-grasping is a fundamental task in robotics; it combines the spatial localization of a graspable object from sensory data and the computation of the inverse kinematics (IK) to move the end effector into a suitable position to grasp an object. Neural learning approaches can offer advantages compared to traditional solutions for grasping: Object localization and computation of the grasp action can be learned end-to-end in a neural architecture. Once trained, the neural network allows robust grasping with fast and constant computation time. Training a network can be approached in two ways: Deep reinforcement learning uses trial and error exploration to gradually improve a policy [1]. While this method is thriving in virtual environments, the long training times can be problematic for physical robots [2]. This constraint is overcome by the second approach, supervised learning, which directly learns from optimally annotated training samples. However, these samples are not readily available.

To this end, we present a supervised neural architecture and an off-line sampling strategy based on prerecorded images, tag localization and classical inverse kinematics in a virtual environment to generate suitable training samples. Samples are annotated independent of the robotic hardware, saving time and preventing possible damage or wear. The approach is evaluated on the industrial 6-DoF UR5 robot arm as it is shown in Figure 1. UR5 reaches for an object on a table. Our results are compared to a related approach on a developmental humanoid robot with human-like joint limits. A critical analysis points to possible challenges when applying neurocognitive models from a developmental robot platform to industrial style manipulators.

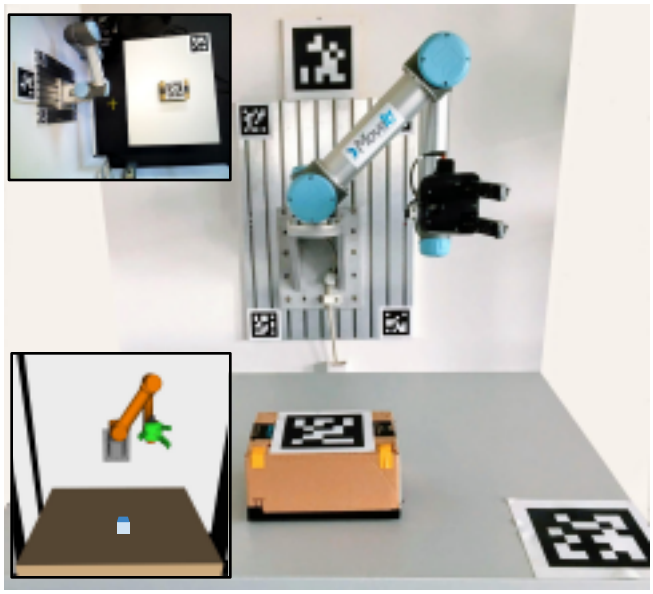

Fig. 1. The UR5 robot arm in the experimental setup with an AprilTag-marked grasp-object. The small picture shows the recreation of the scene in the Rviz simulation environment. The AprilTag on the object is used to locate it in the scene. A traditional IK-solver is used to find a suitable joint configuration for grasping in the simulation to generate training samples.

II. BACKGROUND: GENERATING SAMPLES FOR (SUPERVISED) NEURAL VISUOMOTOR LEARNING

Reaching for grasp requires to locate the desired object in a scene, often using visual sensors, and then computing the inverse kinematics and to move the end-effector into a grasping position. Deep neural architectures can solve both tasks in a single processing step, i.e., when feeding an image of a scene, the network can output a joint configuration that moves the end effector into a suitable grasping position.

For training such a visuomotor neural architecture, samples are needed that associate the visual input (an image of an object in the environment) with the correct joint values to reach for grasping. These samples can be generated in various ways: Levine et al. [3] exploit the known forward kinematics of a secondary robot arm to move an object through space. While this method does not directly supply the joint configuration to grasp the object, it enables the training to take advantage of the full state of the scene during policy learning. Kerzel and Wermter [4] invert the

*The authors gratefully acknowledge partial support from the German Research Foundation DFG under project CML (TRR 169) and the European Union under project SECURE (No 642667).

[1]Knowledge Technology, [2]TAMS, Department of Informatics, University of Hamburg, Germany. 6beik, kerzel, zamani, goerner, eppe, wermter @informatik.uni-hamburg.de

grasping task into a placement task that can autonomously be executed by a humanoid robot in a self-learning cycle and use the generated samples to train a neural network. This method, however, relies on an initial manual motor training of the robot. Hindsight Experience Replay, introduced by Andrychowicz et al. [5], applies a post hoc manipulation to random exploration samples during deep reinforcement learning to turn them into optimal samples.

We extend the state-of-the-art with a novel sampling method that utilizes a traditional inverse kinematics solver in a simulation environment to generate correct joint value annotations for images taken in a real environment. The samples are used to train the top layers of a pre-trained convolutional neural network architecture, which further reduces training time and increases sample efficiency.

III. Method and Experimental Setup

In our experimental setup, we use an industrial UR5[1] arm to reach for a small box on a table. The UR5 has six rotational degrees of freedom with a working radius of 850 mm. Each joint has a full joint range of $[-2\pi, +2\pi]$ radian, but is further restricted to $[-\pi, +\pi]$ in practice. In the case of fixed end-effector orientation the IK-solver produces 8 different solutions per target in which we reduced it to one which is the closest to previous one. This approach keeps the solutions similar to each other and makes the prediction easier for the network. The target object is a box marked with an AprilTag for accurate localization, and a ceiling camera to capture the top surface of the table and the tag easily.

The learning consists of three phases: in the first phase, a box is placed in various positions on the table and images are recorded, as shown in figure 1. In the second phase, the recorded images are annotated with a suitable joint configuration to grasp the object in the depicted position using tag-based localization and inverse kinematics in a virtual environment. In the third phase, the neural network is trained with the annotated samples.

A. Sample Collection and Annotation

Initially, four thousand images of the grasp-object at 400 different position on the table are collected. Figure 2 shows the distribution of the object positions in a regular grid. We use different robotic tools to annotate the collected samples with a joint configuration suitable for grasping in a safe and reliable way, independent of real hardware. The setup, including the robot, the table, and the object to be grasped were recreated in a MoveIt! [6] planning environment for collision-aware IK solving. For each collected image, the exact position of the box was computed using the AprilTag on the object and a reference tag on the wall. The TracIK solver for MoveIt! [7] was then used to compute joint configurations suitable to grasp the object. The final position for the grasp was added as an annotation to the training images. For all 4.000 recorded images, a suitable grasp position was found.

[1] https://www.universal-robots.com/products/ur5-robot/

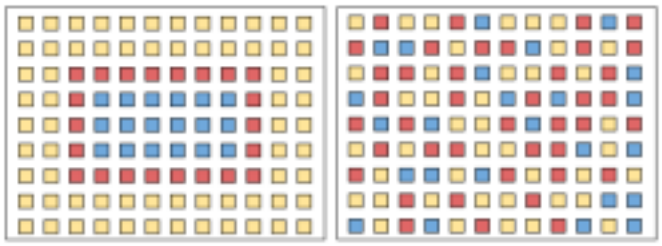

Fig. 2. The arrangement of the two different datasets. Yellow squares indicate the position of training instances, red squares indicate validation instances and blue squares represent test instances. The left *extrapolation* training set proved to be harder for neural learning than the right *interpolation* dataset.

B. End-to-End Visuomotor Neural Architecture

Our neural architecture is adapted from Kerzel and Wermter [4]; it consists of a convolutional part that takes a raw image as input (224 x 224 pixels), followed by four dense layers and an output layer for the six joint values. We evaluate both the use of a randomly initialized stack of convolution and pooling layers (based a simplified VGG-16 architecture) as well as a pre-trained VGG-16 vision network [8]. The structure of the pre-trained network is shown in Figure 3.

Finally, the network is trained with fully annotated samples consisting of an image, showing the object on the table, and a suitable joint configuration to grasp the object.

The network is trained with the Adam optimizer at a learning rate of 10^{-4}. All non-pre-trained layers are initialized with Glorot uniform; all layers use Rectified Linear Unit (ReLU) as the activation function. The overall network architecture was empirically determined and optimized. The architecture was implemented in Keras [9] with the Tensorflow backend [10] using a pre-trained VGG16 network.

IV. Results and Discussion

To evaluate the sampling method and neural reach-for-grasp learning, two different training sets were designed. For the *interpolation* training set, a split between 60% train, 20% validation and 20% test data was performed, resulting in neighboring train and test items. In the *extrapolation* test set, the training data was selected from the edges of the table, while validation and test data were selected from the middle of the table, forcing the neural network to extrapolate from dissimilar data. Figure 2 illustrates the spatial distribution of the different datasets.

Figure 4 shows the resulting learning curves (validation loss) for the two test sets (*interpolation* and *extrapolation*) and the two neural architectures (*pre-trained* and *non-pre-trained*). The curves show the mean squared error for each joint (in radian) from three repeated training runs. As can be seen, for both architectures the *extrapolation* test set yields higher errors. This can easily be explained by the fact that the extrapolation of fully unknown positions from a training set of positions that are clearly different is a very hard task for a neural network. In contrast, for the *interpolation* data set, the network's ability to generalize from the training data to the

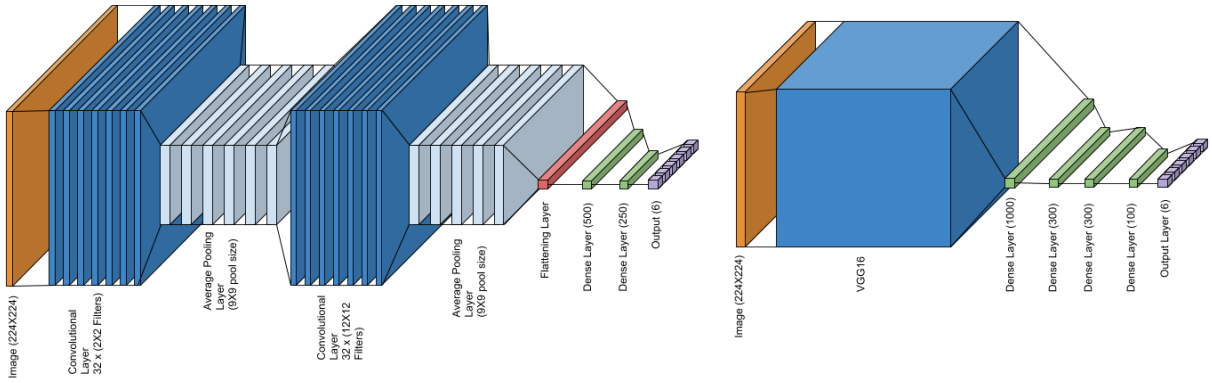

Fig. 3. Neural architecture. (left) Architecture based on the simplified VGG-16 network. (right) Architecture based on the pre-trained VGG-16 network.

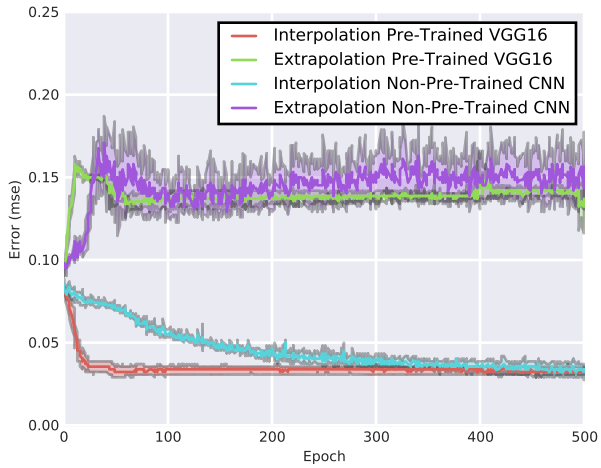

Fig. 4. Validation loss over 500 training epochs for all four conditions: *interpolation* and *extrapolation* with the *non-pre-trained CNN* architecture and the *pre-trained VGG16* architecture

closely related validation data leads to a relatively low loss. As expected, for both training sets the pre-trained networks converge faster and show a more stable performance with less variance. The final test errors for all conditions show a similar distribution as the validation losses. For the *pre-trained, interpolation* condition the MSE is 0.078, for the *non-pre-trained-interpolation* it is 0.067. The MSE indicates the error in each joint in radian.

Overall, the method is working as expected but is not on par with traditional IK and object localization methods and also falls behind when compared to a related approach on a developmental humanoid robot [4]. The higher error can be explained by the typical challenges for neural networks: The larger joint space of the UR5 robotic arm compared to the significantly more constrained humanoid robot amplifies errors of the network in a linear way. These errors, in turn, can in part be attributed to redundant but inconsistent training samples in which spatially close or identical positions are reached for with (vastly) different joint configurations. We conclude that, avoiding redundant samples and constraining joint values as much as possible can help to minimize errors in neural visuomotor learning.

V. Conclusion

We have presented a method for off-line generation of samples for a neural end-to-end learning approach on an industrial 6-DoF robot arm using established tools from the robotics community, like state-of-the-art inverse kinematics solvers and tag-based object localization. We evaluated the learning results with different neural architectures and could show that using pre-trained vision components further increases the sample efficiency. Thus, we contribute to the possibility to apply neural learning methods on robotic hardware while minimizing physical training times.

We also identify possible challenges when combining classical robotics methods and neural learning approaches like the neural network's sensitivity to redundant solutions. In future work, we aim to overcome these challenges with more robust neural visuomotor architectures and improved sampling strategies.

References

[1] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[2] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 3406–3413.

[3] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, no. 39, pp. 1–40, 2016.

[4] M. Kerzel and S. Wermter, "Neural end-to-end self-learning of visuomotor skills by environment interaction," in *International Conference on Artificial Neural Networks (ICANN)*, 2017 accepted.

[5] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba, "Hindsight experience replay," *CoRR*, vol. abs/1707.01495, 2017. [Online]. Available: http://arxiv.org/abs/1707.01495

[6] S. Chitta, I. Sucan, and S. Cousins, "Moveit![ros topics]," vol. 19, pp. 18–19, 03 2012.

[7] P. Beeson and B. Ames, "TRAC-IK: An open-source library for improved solving of generic inverse kinematics," in *Proceedings of the IEEE RAS Humanoids Conference*, Seoul, Korea, November 2015.

[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[9] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[10] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.