Action-Conditioned Convolutional Future Regression Models for Robot Imitation Learning

Alan Wu*, AJ Piergiovanni*, and Michael S. Ryoo
School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN 47408
mryoo@indiana.edu

1. Introduction

Based on what is seen (i.e. visual input), humans are able to visually predict (i.e. regress) what the scene will look like after taking a certain action. Further, humans are able to take advantage of such predictions to select optimal actions for the task they are working on. Using example videos, robots can also learn to visually imagine the future consequence of taking an action. This can be viewed as learning a function mapping a raw image frame (conditioned on a particular action) to the future image frame. Once learned, the future regression function can be combined with an action policy learning framework (e.g. reinforcement or imitation learning), enabling better robot action learning for given tasks.

We formulate the problem of robot action learning as the learning of convolutional neural network (CNN) model parameters. Prior works were limited to the forecasting of small pixel motion in a static robot video [1] or were done without considering multiple action possibilities [2]. Our future frame regression is designed to handle entire scene changes conditioned on input action selected from multiple candidates, enabling the network to learn representations to predict large scene changes. We compare several CNN architectures designed for the scene-level future regression (Fig. 1), investigating the better function modeling strategies for robot *imitation learning*. Given a set of expert example videos as training data, our robot simultaneously learns the action-conditioned future regression function and a CNN-based value function to select the optimal action for a given task. We compare our future regression models with the standard behavioral cloning and Q function learning.

We use a ground mobility robot for our real-time experiments, which were performed in a real-world environment with everyday objects (e.g. desks, chairs, cabinets) without any markers, in order to compare our different CNN models for future regression and imitation learning. Importantly, we do not provide any explicit label showing what the target looks like, or allow the robot to use any localization technique when making the action decisions. Our results confirm the benefits of action-conditioned future regression.

*These authors contributed equally to the paper.

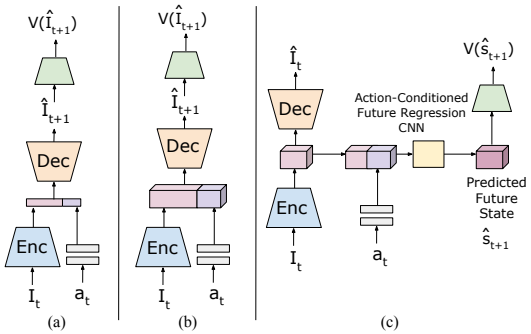

Figure 1: Action-conditioned future scene regression with (a) linear action representation and (b) convolutional action representation. We also tried (c) future 'representation' regression, to be combined with the value function.

2. Approach

We present different CNN architectures for robot action learning. Given an image input at each time step, our goal is to make the robot decide its optimal action to reach the target object. In imitation learning, this is done by learning an action model from human expert training data.

In its simplest form, this can be formulated as the learning of the action function f_w that produces the robot action a_t based on the robot's visual input I_t: $a_t = f_w(I_t)$ where a_t can be the direct motor control commands or more abstract actions. Direct learning of such function parameter w (from supervised training data) is often called 'behavioral cloning.' CNNs are good function approximators that can be used to approximate f_w. Alternatively, we can also use the Q function learning to choose the optimal action. The Q function evaluates the goodness of state-action pairs, as also was done in Deep Q learning [3]. The only difference here is that this function is being trained solely based on expert trajectories. The action is decided by $a_t = \arg\max_a Q(s_t, a)$ where s_t is often I_t.

2.1. Action-conditioned future regression

First, in order to obtain a better state representation, we propose to take advantage of an action-conditioned autoencoder for future regression. This model, shown in Fig. 1(a), is composed of two CNNs (i.e. two functions): the encoder Enc and the decoder Dec. The encoder maps the current image to a latent linear representation, $z_I = Enc(I_t)$. We also use a neural network, Act, to learn a representation of the action, $z_a = Act(a_t)$. The decoder then reconstructs the future image from this representation.

$$\hat{I}_{t+1} = Dec([Enc(I_t), Act(a_t)]). \quad (1)$$

This model, trained to directly minimize L_1 error between the predicted image and the ground truth image, is further combined with another CNN to estimate the value function.

We also extend our approach as described in Fig. 1(b) to preserve spatial information by doing the convolutional future regression while concatenating the intermediate feature map with the learned action representation also having spatial dimensionality. Lastly, similar to [2], we learn future regression directly in the representation space (Fig. 1(c)). We first train a denoising autoencoder to reconstruct a given image. The latent space, $z_I = Enc(I_t)$, is our state. We then train a neural network to learn a representation of the action $z_a = Act(a_t)$. These representations are concatenated together, followed by a future regression CNN, R, to predict the next state:

$$\hat{s}_{t+1} = R([Enc(I_t), Act(a_t)]) \quad (2)$$

This network has 3 loss functions: image reconstruction, future state regression, and value function learning.

3. Experiments

We perform both offline evaluation of the robot action model as well as online evaluation of real-time robot experiments. We collected a set of 10 random exploration trajectories for an average of 1750 steps to learn the environment. In addition, a set of expert trajectories with 7 different target objects was collected. The target location and the initial robot placement were varied. For each trajectory, we only annotate the frame with the current robot pose. On average, we collected 36 110-frame trajectories for each of the 7 target objects. We split our dataset to 32 training trajectories and 4 test trajectories per target.

3.1. Evaluation of our models

To test the potential of the various models, we conducted a set of experiments using the held-out expert trajectories in our dataset. Table 1 shows the results evaluating our models on the unseen trajectories. We show mean linear and rotational errors between selected actions by each network and optimal actions. Future scene regression models yielded the best performance.

Table 1: Evaluation of our various models on the held-out expert trajectories.

Model	Error cm	Error deg
Behavioral Cloning	21.98	25.23°
CNN Q Function Learning	9.91	8.58°
Future Scene Regression (linear action)	**8.46**	9.74°
Future Scene Regression (conv action)	9.06	**8.35**°
Future State Regression (conv action)	11.18	21.70°

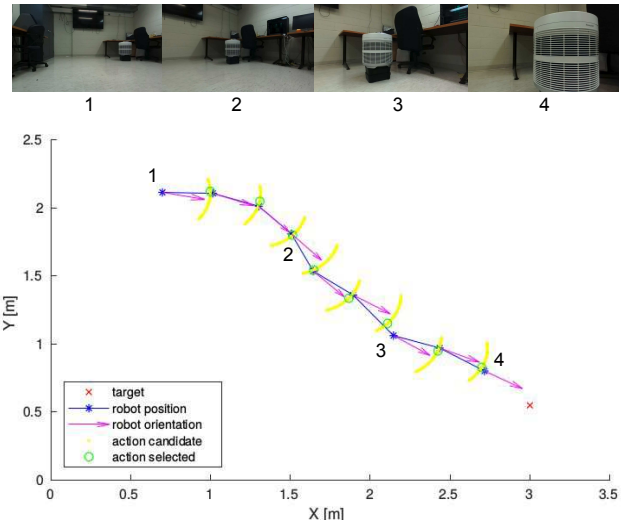

Figure 2: Sample trajectory.

3.2. Real-time robot experiments

We conducted a set of real-time experiments with a ground mobility robot in a complex real-world environment to illustrate the implementation of the action network models we learned. For each model, we ran 18 trials each for two different target objects. A successful run entails fewer than 20 steps to reach within 0.5 meters of the target and to capture an image of the target within the 0.5m. Fig. 2 shows a trajectory of an example run, including frames our robot obtains and processes during its task. It is important to note that we do not explicitly perform localization of the target within the image. Rather, we rely strictly on the network to select the optimal action based on visual input. In Table 2, we see that the convolutional future scene regression (Fig. 1(d)), because of its superior ability to preserve spatial information, yielded the best results.

Table 2: Real-time robot experiment results reporting the task success rate.

Model	Target 1	Target 2	Average
CNN Q Function Learning	50%	44%	47%
Linear Future Scene Regression	89%	67%	78%
Conv Future Scene Regression	**100%**	**94%**	**97%**

References

[1] C. Finn and S. Levine. Deep visual foresight for planning robot motion. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017. 1

[2] J. Lee and M. S. Ryoo. Learning robot activities from first-person human videos using convolutional future regression. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017. 1, 2

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015. 1