Visionary: Neural Architecture Search for Robot Learning

Iretiayo Akinola[1,2*], Anelia Angelova[1], Yao Lu[1], Yevgen Chebotar[1], Dmitry Kalashnikov[1]
Jacob Varley[1], Julian Ibarz[1], Michael S. Ryoo[1,3]
[1]Robotics at Google
[2]Columbia University
[3]Stony Brook University
mryoo@google.com

Abstract

We propose a vision-based architecture search algorithm for robot manipulation learning, which discovers interactions between low dimension action inputs and high dimensional visual inputs. Our approach automatically designs architectures while training on the task – discovering novel ways of combining and attending image feature representations with actions *as well as features from previous layers. The obtained new architectures demonstrate better task success rates, in some cases with a large margin, compared to a recent high performing baseline. Our real robot experiments also confirm that it improves grasping performance by 6%. This is the first approach to demonstrate a successful neural architecture search and attention connectivity search for a real-robot task.*

This is a two-page extended abstract. The full version of the paper could be found at [1]

1. Introduction

For many decades, autonomous robots have been effective in structured environments, such as factories. However, creating robots that work in less-structured environments e.g., households, offices and warehouses, remains challenging and is one of the main goals for autonomous robotics. To operate in diverse and complex environments, autonomous robots have to be intelligent, adaptable, and able to learn from their sensory observations and experience. Recent progress in machine learning research has enabled robotic agents to acquire policies that can map sensory observations (i.e., images) to intelligent actions. For example, large progress has been shown in grasping by learning from raw visual inputs in an end-to-end manner [6, 10, 14, 3, 13, 16, 8, 9, 17].

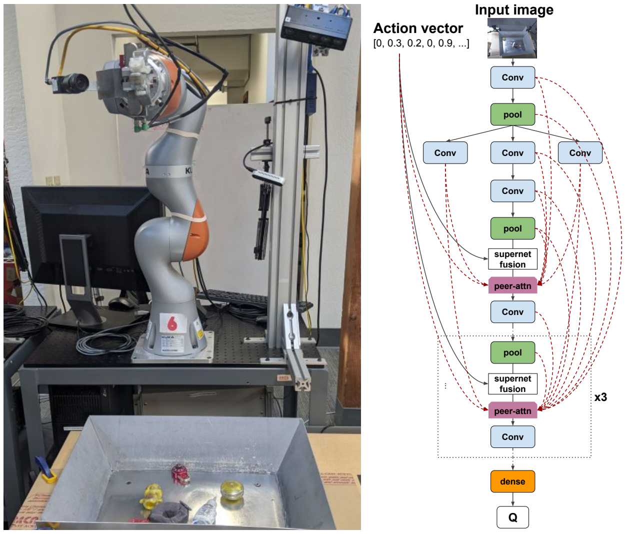

Figure 1. We develop a fully differentiable visual architecture search method that can automatically learn a superior architecture while training to adapt to the task on hand. We deploy our approach on a real-robot grasping task.

Despite these recent breakthroughs, existing methods still have limitations in terms of adaptability to different environments, task complexity, sample complexity among others. Getting a robot to perform long horizon tasks like packaging in a way that generalizes well to different setting by learning from minimal experiences is still challenging for a number of reasons. One challenge is designing the visual models which are best suited for performing complex robotic tasks. This is often done by using a network designed for another task, typically borrowing from standard computer vision tasks, such as classification, or by manual design of the architecture. Furthermore, robot learning happens in a tight vision-action-control feedback loop and the action taken at each step is a crucial input. Previous work in robotics have recognized that in addition to feature representation, action representation matters significantly in robotics manipulation [5], but is not clear how these two

[*] Work done during an internship at Google.

conceptually different sources of data should be combined.

In this work we propose a novel differentiable architecture search that jointly learns the model architecture while training the model itself for the task at hand. The idea is to make the search explore how the action and other low-dimensional sensory inputs interact and merge with features from high dimensional visual input, as this affects how well the network captures the interplay between action and the other inputs for accomplishing complex tasks. This is done within Reinforcement Learning (RL)-based robot learning context, where in addition to learning the main architecture to generate visual features and combine them with action inputs, the robot is learning a policy to accomplish the manipulation task.

Our proposed algorithm, which we term *Visionary*, leads to the discovery of new and successful architectures, exploring the combination of action and visual inputs quickly and efficiently via a differentiable one-shot architecture search.

2. Approach

Reinforcement Learning (RL) formulation: We use a variant of Q-Learning algorithm for continuous actions (QT-Opt [7]) to learn state-action value function. This Q-value function, given as $\mathcal{Q}^{\pi}(s,a) = r(s,a) + \gamma \max_{a'} \mathcal{Q}^{\pi}(s',a')$, captures the discounted sum of rewards that the agent can accumulate starting from a state s, executing action a and following the policy π. s' is the next state and γ is a discount factor. The policy $\pi(s)$ that gives the action for each state is obtained by optimizing the learned Q-function using the cross-entropy method – a derivative-free optimization algorithm.

Search formulation: In addition to learning the weights of a Q-Value function for vision-based control, we learn the neural architecture itself to accomplish the task more successfully. Instead of tackling this problem with the standard bi-level optimization, we take an alternative strategy: one-shot differentiable joint optimization [12]. Our differentiable architecture discovery formulation allows directly regressing optimal architecture parameters jointly with the CNN filter values during the RL training. This enables plugging in our approach into any existing reinforcement learning framework.

Architecture search: We search for action merging types/locations as well as attention connectivity.

The action merging is found by building a 'supernet' module and inserting it at possible merging locations. The supernet's action-merging (i.e., fusion) module consists of a number of action merging blocks f_j, each of which takes in an intermediate representation tensor x and the action vector a as an input. The output of the module is given as: $x_{i+1} = f_j(x_i, a)$ where x_i is the intermediate representation at the ith layer of the CNN model while each action merging operation f_j is applied and combined to obtain a new representation x_{i+1}. We learn a set of weights w_{ij} (at each level i), corresponding to each candidate operation f_j. The output of the supernet module F_i, given the input x and a is computed as:

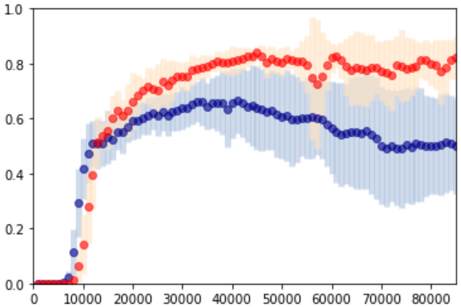

Figure 2. Mean and standard deviation of our model (red) compared to the baseline (blue) at every 1K time steps. The X-axis is the training iterations, and the Y-axis is the task success rate.

$$F_i(x,a) = \sum_j \sigma(w_{ij}) \cdot f_j(x,a) \tag{1}$$

where $\sigma(w_{ij})$ is formulated with a softmax function: $\sigma(w_{ij}) = e^{w_{ij}} / \sum_k e^{w_{ki}}$ The supernet module computes a weighted summation of all possible operations while constraining the weights with softmax, thereby making a soft-selection of one of the operations. Once these weights are learned and finalized, non-argmax connections are pruned, making it spend the same amount of compute to conventional action merging during the inference.

For the attention connectivity search, we formulate our peer-attention module [15] similar to the supernet module, so that the optimal connection is found through the gradient computation of differentiable parameters. Let the function $G_i(x_i)$ denote the peer-attention module:

$$G_i(x_i) = \sum_{v \in \{a, x_1, \cdots, x_i\}} \text{Attn}\big(x_i,\ h_{iv} \cdot \text{GAP}(v)\big) \tag{2}$$

where h_{iv} is a scalar weight corresponding to the representation v (constrained with sigmoid), GAP$(\cdot)$ is a (spatial) global average pooling layer, and Attn$(\cdot)$ is the attention function often implemented as a broadcasted element-wise multiplication. This allows the model to explore various connections between the representations.

3. Experimental Results

Our experimentation platform for learning is adapted from QT-Opt [7]. We develop and test our approach in simulation (using the PyBullet-based physics simulator [4]). We also validate our results on the grasping task using real hardware. We use an RGB image of size 472x472 from a static camera mounted above the workspace over the shoulder of the robot arm. The action space consists of a gripper pose displacement, and an open/close command.

Figure 2 shows performance of the proposed approach compared to the baseline (from [7, 2]) on the block stacking task in simulation. We further test our approach on a KUKA Robot for the task of grasping random objects from a bin. Using offline real robot data [11], the model architecture is discovered and learned. It is then tested on the real-time KUKA robot in a different environment, which is more challenging and mimics setups in a warehouse or a factory floor. The performances are: Base model (from [7]) 64.6 and Visionary (ours) 70.8.

References

[1] Iretiayo Akinola, Anelia Angelova, Yao Lu, Yevgen Chebotar, Dmitry Kalashnikov, Jacob Varley, Julian Ibarz, and Michael S. Ryoo. Visionary: Vision architecture discovery for robot learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021. 1

[2] Iretiayo Akinola, Jacob Varley, and Dmitry Kalashnikov. Learning precise 3d manipulation from multiple uncalibrated cameras. In *ICRA*, 2020. 3

[3] Cesar Cadena, Anthony Dick, and Ian D. Reid. Multi-modal auto-encoders as joint estimators for robotics scene understanding. In *Robotics Science and Systems*, 2016. 1

[4] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. 2016. 2

[5] Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Learning visual feature spaces for robotic manipulation with deep spatial autoencoders. *arXiv preprint arXiv:1509.06113*, 25, 2015. 1

[6] M. Gualtieri, A. ten Pas, K. Saenko, and R. Platt. High precision grasp pose detection in dense clutter. In *IROS*, 2015. 1

[7] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. In *CoRL*, 2018. 2, 3

[8] Torsten Kröger Lars Berscheid, Thomas Rühr. Improving data efficiency of self-supervised learning for robotic grasping. In *ICRA*, 2019. 1

[9] Michelle A Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019. 1

[10] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. In *IJRR*, 2015. 1

[11] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020. 3

[12] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture seach. In *ICLR*, 2019. 2

[13] 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. In *RSS*, 2017. 1

[14] Joseph Redmon and Anelia Angelova. Real-time real-time grasp detection using convolutional neural networks. In *ICRA*, 2015. 1

[15] Michael S. Ryoo, AJ Piergiovanni, Juhana Kangaspunta, and Anelia Angelova. AssembleNet++: Assembling modality representations via attention connections. In *ECCV*, 2020. 2

[16] Yu Xiang, Tanner, Schmidt, andenkatraman Narayanan, , and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Robotics: Science and Systems (RSS)*, 2018. 1

[17] Christopher Xie, Yu Xiang, Arsalan Mousavian, and Dieter Fox. The best of both modes: Separately leveraging rgb and depth for unseen object instance segmentation. In *CoRL*, 2019. 1