Tuning Modular Networks with Weighted Losses for Hand-Eye Coordination

Fangyi Zhang, Jürgen Leitner, Michael Milford, Peter I. Corke *†

Abstract

This paper introduces an end-to-end fine-tuning method to improve hand-eye coordination in modular deep visuo-motor policies (modular networks) where each module is trained independently. Benefiting from weighted losses, the fine-tuning method significantly improves the performance of the policies for a robotic planar reaching task.

1. Introduction

Recent work has demonstrated robotic tasks based directly on real image data using deep learning, for example robotic grasping [2]. However these methods require large-scale real-world datasets, which are expensive, slow to acquire and limit the general applicability of the approach.

To reduce the cost of real dataset collection, we used simulation to learn robotic planar reaching skills using the DeepMind DQN [3]. The DQN showed impressive results in simulation, but exhibited brittleness when transferred to a real robot and camera [4]. By introducing a bottleneck to separate the DQN into perception and control modules for independent training, the skills learned in simulation (Fig. 1A) were easily adapted to real scenarios (Fig. 1B) by using just 1418 real-world images [5].

However, there is still a performance drop compared to the control module network with ideal perception. To reduce the performance drop, we propose fine-tuning the combined network to improve hand-eye coordination. Preliminary studies show that a naive fine-tuning using Q-learning does not give the desired result [5]. To tackle the problem, we introduce a novel end-to-end fine-tuning method using weighted losses in this work, which significantly improved the performance of the combined network.

2. Methodology

We consider the planar reaching task, which is defined as controlling a 3 DoF robot arm (Baxter robot's left arm) so that in operational space its end-effector position $\mathbf{x} \in \mathbb{R}^2$ moves to the position of the target $\mathbf{x}^*$ in a vertical plane (ignoring orientation). The reaching controller adjusts the robot configuration (joint angles $\mathbf{q} \in \mathbb{R}^3$) to minimize the error between the robot's current and target position, i.e., $\|\mathbf{x} - \mathbf{x}^*\|$. At each time step 1 of 9 possible actions $a \in \mathbf{a}$ is chosen to change the robot configuration: 3 per joint – increasing or decreasing by a constant amount (0.04 rad) or leaving it unchanged. An agent is required to learn to reach using only raw-pixel visual inputs I from a monocular camera and their accompanying rewards r.

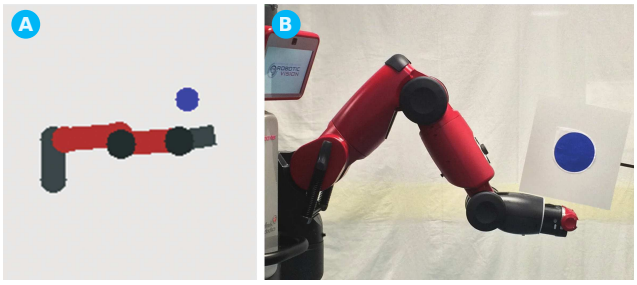

Figure 1. A technique to improve hand-eye coordination for better performance when transferring deep visuo-motor policies for a planar reaching task from simulated (A) to real environments (B).

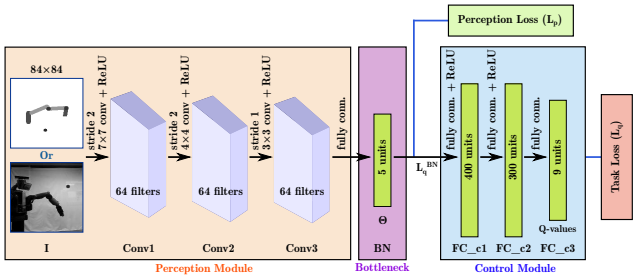

Figure 2. A modular neural network is used to predict Q-values given some raw pixel inputs. It is composed of perception and control modules. The perception module which consists of three convolutional layers and a FC layer, extracts the physically relevant information ($\boldsymbol{\Theta}$ in the bottleneck) from a single image. The control module predicts action Q-values given $\boldsymbol{\Theta}$. The action with a maximum Q-value is executed. The architecture is similar to that in [5], but has an additional end-to-end fine-tuning process using weighted perception and task losses. Note that the values in $\boldsymbol{\Theta}$ are normalized to the interval $[0, 1]$.

The network has the same architecture and training method to [5], but with an additional end-to-end fine-tuning using weighted losses, as shown in Fig. 2. The perception network is first trained to estimate the scene configuration $\boldsymbol{\Theta} = [\mathbf{x}^* \, \mathbf{q}] \in \mathbb{R}^5$ from a raw-pixel image I using the quadratic loss function

$$L_p = \frac{1}{2m} \sum_{j=1}^{m} \left\| y(I^j) - \boldsymbol{\Theta}^j \right\|^2,$$

*FZ, JL, MM, PIC are with the Australian Centre for Robotic Vision (ACRV), Queensland University of Technology (QUT), Brisbane, Australia. fangyi.zhang@hdr.qut.edu.au

†This research was conducted by the Australian Research Council Centre of Excellence for Robotic Vision (project number CE140100016). Additional computational resources and services were provided by the HPC and Research Support Group at QUT.

where $y(I^j)$ is the prediction of $\mathbf{\Theta}^j$ for I^j; m is the number of samples. The control network is trained using K-GPS [5] where network weights are updated using the Bellman equation which is equivalent to the loss function

$$L_q = \frac{1}{2m}\sum_{j=1}^{m}\left\| Q(\mathbf{\Theta}_t^j, a_t^j) - (r_t^j + \gamma \max_{a_{t+1}^j} Q(\mathbf{\Theta}_{t+1}^j, a_{t+1}^j))\right\|^2,$$

where $Q(\mathbf{\Theta}_t^j, a_t^j)$ is the sum of future expected rewards $\sum_{k=0}^{\infty}\gamma^k r_{t+k}^j$ when taking action a_t^j in state $\mathbf{\Theta}_t^j$. γ is a discount factor applied to future rewards.

After separate training for perception and control individually, an end-to-end fine-tuning is conducted for the combined network (perception + control) using weighted task (L_q) and perception (L_p) losses. The control network is updated using only L_q, while the perception network is updated using the weighted loss

$$L = \beta L_p + (1-\beta)L_q^{BN},$$

where L_q^{BN} is a pseudo-loss which reflects the loss of L_q in the bottleneck (BN); $\beta \in [0, 1]$ is a balancing weight. From the backpropagation algorithm [1], we can infer that $\delta_L = \beta\delta_{L_p} + (1-\beta)\delta_{L_q^{BN}}$, where δ_L is the gradients resulted by L; δ_{L_p} and $\delta_{L_q^{BN}}$ are the gradients resulting respectively from L_p and L_q^{BN} (equivalent to that resulting from L_q in the perception module).

3. Experiments and Results

We evaluated the feasibility of the proposed approach using the metrics of Euclidean distance error d (between the end-effector and target) and average accumulated reward $\bar{R}$ (a bigger accumulated reward means a faster and closer reaching to a target) in 400 simulated trials. For comparison, we evaluated three networks: **Initial**, **Fine-tuned** and **CR**. **Initial** is a combined network without end-to-end fine-tuning, which is labelled as EE2 in [5] (comprising **FT75** and **CR**). **FT75** and **CR** are the selected perception and control modules which have the best performance individually. **Fine-tuned** is obtained by fine-tuning **Initial** using the proposed approach. **CR** works as a baseline indicating performance upper-limit.

In fine-tuning, $\beta = 0.8$, we used a learning rate between 0.01 and 0.001, a mini-batch size of 64 and 256 for task and perception losses respectively, and an exploration possibility of 0.1 for K-GPS. These parameters were empirically selected. To make sure that the perception module remembers the skills for both simulated and real scenarios, the 1418 real samples were also used to obtain δ_{L_p}. Similar to **FT75**, 75% samples in a mini-batch were from real scenarios, i.e., at each weight updating step, 192 extra real samples were used in addition to the 64 simulated samples in the mini-batch for δ_{L_q}.

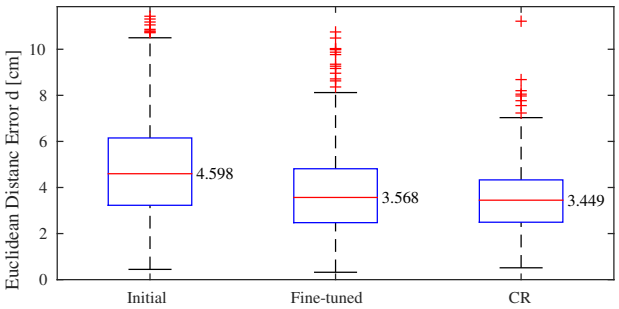

Figure 3. The box-plots of distance errors of different networks, with median values displayed. The crosses represent outliers.

Table 1. Planar Reaching Performance

Nets	d_{med}		d_{Q3}		$\bar{R}$
	[cm]	**[pixels]**	**[cm]**	**[pixels]**	**[\]**
Initial	4.598	1.929	6.150	2.581	0.319
Fine-tuned	3.568	1.497	4.813	2.020	0.626
CR	3.449	1.447	4.330	1.817	0.761

Results are summarized in Fig. 3 and Table 1. d_{med} and d_{Q3} are the median and third quartile of d. The error distance in pixels in the 84×84 input image is also listed. We can see that **Fine-tuned** achieved a much better performance (22.4% smaller d_{med} and 96.2% bigger $\bar{R}$) than **Initial**. The fine-tuned performance is even very close to that of the control module (**CR**) which controls the arm using ground-truth $\mathbf{\Theta}$ as sensing inputs. We also did the same evaluations in 20 real-world trials on Baxter, and achieved similar results.

The experimental results show the feasibility of the proposed fine-tuning approach. Improved hand-eye coordination in modular deep visuo-motor policies is possible due to fine-tuning with weighted losses. The adaptation to real scenarios can still be kept by presenting (a mix of simulated and) real samples to compute the perception loss.

References

[1] Y. LeCun. A theoretical framework for back-propagation. In D. Touretzky, G. Hinton, and T. Sejnowski, editors, *Proceedings of the 1988 Connectionist Models Summer School*, pages 21–28, CMU, Pittsburgh, Pa, 1988. Morgan Kaufmann.

[2] S. Levine, P. P. Sampedro, A. Krizhevsky, and D. Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. In *International Symposium on Experimental Robotics (ISER)*, 2016.

[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):529–533, 2015.

[4] F. Zhang, J. Leitner, M. Milford, B. Upcroft, and P. Corke. Towards vision-based deep reinforcement learning for robotic motion control. In *Australasian Conference on Robotics and Automation (ACRA)*, 2015.

[5] F. Zhang, J. Leitner, B. Upcroft, and P. Corke. Transferring vision-based robotic reaching skills from simulation to real world. Technical report, 2017.