Which Kernels to Transfer in Deep Q-Networks?

Anonymous Author(s)
Affiliation
Address
email

1 Introduction

Deep Learning (DL) approaches have shown good results in complex tasks, however, they usually require a large number of examples and long training times. Deep Reinforcement Learning (DRL) solves Reinforcement Learning (RL) tasks by taking advantage of DL (using as input raw data and automatically extracting useful features). Although very effective, DRL inherits DL's limitations. Transfer Learning (TL) is an alternative to reduce the training time of DL models by selecting a source model to be used for learning a new target task.

Most of the proposed TL methods in DRL use a source model to train a new one with lower parameters for the same task [1–4], but they do not transfer knowledge to a new task. Other works transfer knowledge to a new task [5–7], but they use an intuitive way for selecting the source task, consequently they do not take into account others that could be more useful than the selected task.

In this paper we describe a method for selecting the best kernels to transfer using the entropy of the output of convolutional layers. The main advantage is that the obtained model for the new task has lower weights to adjust. We tested our approach in Atari games, that is one of the most used benchmarks for DRL. Experimental results show that in some games the proposed method is able to outperform the Deep Q-Networks (DQNs) without transfer.

2 Kernel Selection Using Entropy

An outline of the proposed method is shown in Figure 1. The method uses a sample of the target task in the pre-trained model of the source task. We discretized the outputs after applying the kernels in the input images and select those with a higher entropy value. The outputs of the kernels are normalized with values $\in [0 - 1]$. We hypothesize that the kernels that produce an output with higher diversity of values could be more useful than those that only produce uniform outputs.

Since we only transfer kernels with higher entropy values, the target model has less parameters to tune than the source model. Consequently, we have lower training times. In our experiments we used only 1/4 of the number of units with respect to the source model in each layer for the target model. Additionally, we propose a distance measure between tasks based on the output of a hidden layer, in this case the flatten layer. The measure evaluates a sample of source and target task frames, and obtains the output of the flatten layer. The euclidean distance between each instance is evaluated and the average of every distance is used as the distance between the source and target tasks.

3 Experimental Results

The experimental results are shown in Tables 1 (values lower than 0 means negative jumpstart) 2, and 3 (values higher than one means higher score that the baseline). The labels of the used games are: BX (Boxing), BR (Breakout), FW (Freeway), PN (Pong), and GP (Gopher). The columns correspond to the source tasks and the rows to the target task. We trained the target task using prioritized experience replay and the Adam optimizer instead of RMSprop. We used the Dopamine-rl [8] library for our

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

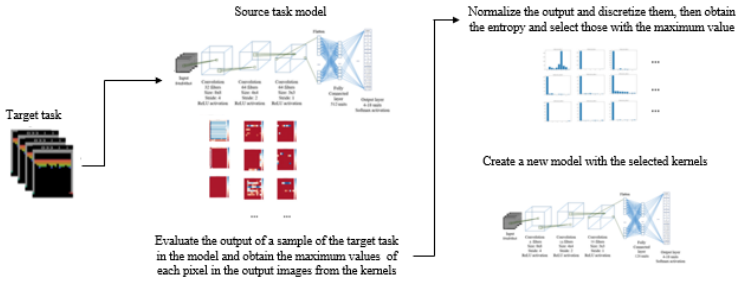


Figure 1: Proposed Method.

experiments. The best model to transfer is within the first two choices selected by the proposed measure (Table 4) for Boxing, Freeway and Gopher.

Table 1: Results of jumpstart.

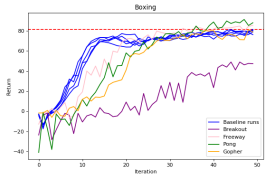
	BX	BR	FW	PN	GP
BX	×	**0.77**	149.37	0.01	-0.99
BR	-6.58	×	-0.99	-1.01	**-0.21**
FW	-3.10	0.41	×	**0.10**	-0.30
PN	-12.13	0.26	335.77	×	-0.42
GP	**-0.44**	0.41	**371.79**	-0.02	×

Table 2: Results of the best policy

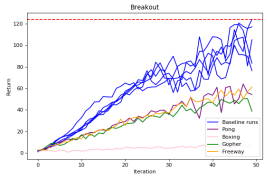
	BX	BR	FW	PN	GP
BX	×	0.06	1.00	0.51	0.00
BR	0.60	×	0.69	0.46	0.81
FW	1.04	0.40	×	**0.88**	0.83
PN	**1.12**	**0.51**	**1.01**	×	**1.04**
GP	0.99	0.40	0.98	0.63	×

Table 3: Results of final policy.

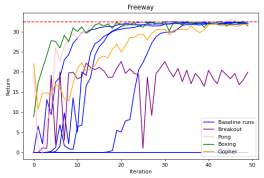
	BX	BR	FW	PN	GP
BX	×	0.05	0.97	0.62	0.00
BR	0.58	×	0.61	0.38	0.50
FW	1.05	0.31	×	0.82	0.77
PN	**1.08**	**0.43**	**0.99**	×	**1.17**
GP	0.96	0.31	0.97	0.70	×



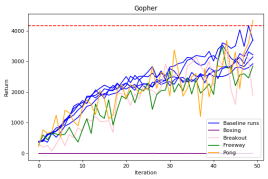
(a) Boxing



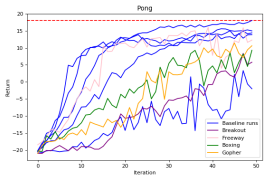
(b) Breakout



(c) Freeway



(d) Gopher



(e) Pong

Table 4: Results of the measure

	BX	BR	FW	PN	GP
BX	×	**0.02**	**0.61**	**1.65**	**0.19**
BR	16.51	×	30.80	31.66	41.74
FW	**2.14**	1.04	×	2.84	1.65
PN	7.27	4.61	6.33	×	1.24
GP	5.35	1.70	1.64	9.55	×

Figure 2: Learning curves for kernel selection approach.

In general we obtain promising results in these initial experiments. Our method outperforms the training times in 3/5 games with the same or even lower number of iterations (in Boxing, Freeway and Gopher). Some games show a positive jumpstart value (Breakout, Freeway and Pong), but only Freeway outperform the best policy. In Boxing and Gopher the proposed method obtained better final accumulated rewards than those obtained without transfer.

Pong and Breakout cases resulted in negative transfer. In Breakout, none of the source task elements were useful for this task, we plan to include more games in the experiments where maybe other pre-trained model could be useful. Pong case is a little different because transfer from Freeway outperform four of five experiments beginning with random weights.

4 Conclusions and Future Work

We propose a method for selecting the best kernels to transfer to a new target task using entropy in DQNs. Our experiments show that our method outperforms in some cases the obtained scores of the baseline method (3/5 games) with only 1/4 of the units in each convolutional layer. For future work, we will test our approach with more Atari games.

References

[1] Simon Schmitt, Jonathan J Hudson, Augustin Zidek, Simon Osindero, Carl Doersch, Wojciech M Czarnecki, Joel Z Leibo, Heinrich Kuttler, Andrew Zisserman, Karen Simonyan, et al. Kickstarting deep reinforcement learning. *arXiv preprint arXiv:1803.03835*, 2018.

[2] Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

[3] Haiyan Yin and Sinno Jialin Pan. Knowledge transfer for deep reinforcement learning with hierarchical experience replay. In *AAAI*, pages 1640–1646, 2017.

[4] Tom Zahavy, Daniel J. Mankowitz, and Shie Mannor. Deep reinforcement learning from expert's experience replay. 2017.

[5] Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning. *arXiv preprint arXiv:1511.06342*, 2016.

[6] Gabriel de la Cruz, Yunshu Du, James Irwin, and Matthew Taylor. Initial progress in transfer for deep reinforcement learning algorithms. 07 2016.

[7] Thomas Carr, Maria Chli, and George Vogiatzis. Domain adaptation for reinforcement learning on the atari. *arXiv preprint arXiv:1812.07452*, 2018.

[8] Pablo Samuel Castro, Subhodeep Moitra, Carles Gelada, Saurabh Kumar, and Marc G. Bellemare. Dopamine: A Research Framework for Deep Reinforcement Learning. 2018.