DCT: Dual Channel Training of Action Embeddings for Reinforcement Learning with Large Discrete Action Spaces

Extended Abstract

Pranavi Pathakota
TCS Research
Mumbai, India
p.pranavi@tcs.com

Hardik Meisheri
TCS Research
Mumbai, India
hardik.meisheri@tcs.com

Harshad Khadilkar
TCS Research
IIT Bombay
Mumbai, India
harshad.khadilkar@tcs.com
harshadk@iitb.ac.in

ABSTRACT

The ability to learn robust policies while generalizing over large discrete action spaces is an open challenge for intelligent systems, especially in noisy environments that face the curse of dimensionality. In this paper, we present a novel framework to efficiently learn action embeddings that simultaneously allow us to reconstruct the original action as well as to predict the expected future state. We describe an encoder-decoder architecture for action embeddings with a dual channel loss that balances between action reconstruction and state prediction accuracy. We use the trained decoder in conjunction with a standard reinforcement learning algorithm that produces actions in the embedding space. Our architecture is able to outperform two competitive baselines in two diverse environments: a 2D maze environment with more than 4000 discrete noisy actions, and a product recommendation task that uses real-world e-commerce transaction data. Empirical results show that the model results in cleaner action embeddings, and the improved representations help learn better policies with earlier convergence.

KEYWORDS

Reinforcement Learning; Self-Supervised learning; Planning and Navigation; Recommender Systems

ACM Reference Format:
Pranavi Pathakota, Hardik Meisheri, and Harshad Khadilkar. 2024. DCT: Dual Channel Training of Action Embeddings for Reinforcement Learning with Large Discrete Action Spaces: Extended Abstract. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

1 INTRODUCTION

Reinforcement learning (RL) has had significant recent success in applications such as games and robotics [7, 10]. However, real-world problems that involve a large number of discrete action choices are still very challenging for traditional RL algorithms. Examples include scenarios such as recommendation systems [1], supply chains [9], complex high fidelity games [2, 13] resource management at scale in data centers [5, 8], investment management [6], where large action spaces are handled indirectly using pre- or post-processing heuristics. The key challenge is with exploring large action spaces sufficiently well to arrive at optimal policies. Furthermore, hand-crafted heuristics for mapping RL outputs to actions become intractable as the number of actions increases.

Recently, the success of state embeddings for complex state spaces has inspired studies on the use of action embeddings along similar lines [4]. The key idea is to learn the RL policy not over raw actions, but over action *representations* in a low dimensional embedding space. If actions with similar effects are grouped close together in the embedding space, the efficiency of exploration is greatly improved. It stands to reason that the better the action representations, the better the chance of reaching good policies.

In this paper, we present an architecture to efficiently learn action embeddings in low dimensional space. We force the embeddings to be rich by imposing the dual task of learning the effect of actions as well as predicting future states. We show experimentally that this helps the RL agents learn better policies in scenarios with large action spaces. We build upon work of Chandak et al. [3] and Pritz et al. [11] and provide a generalized framework for learning embeddings which is not only efficient in encoding transition dynamics between states but also helps in decoding those actions (Fig. 1).

The main contributions of our work are as follows:

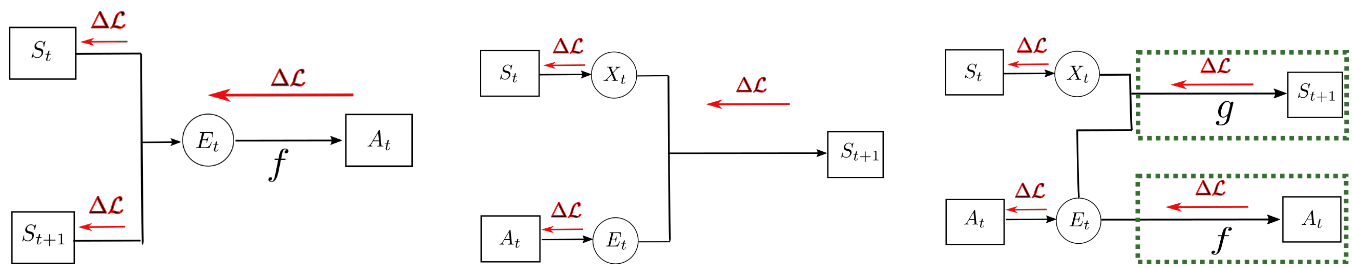
- We propose a new architecture for an action encoder-decoder model which results in a better representation of action embeddings by jointly training encoder and decoder for action reconstruction and next state prediction.
- We present extensive experimentation over a *noisy* maze environment with up to 2^{12} unique actuator actions to validate our model and compare it with previous work and a traditional off policy RL algorithm (DQN).
- We also demonstrate the effectiveness of our algorithm in recommender systems, outperforming baselines on a real-world fashion e-commerce dataset.

2 METHODOLOGY

In this section, we propose a model to efficiently learn action embeddings using an encoder-decoder architecture with Dual Channel Training (DCT). We focus on the explanation of action embeddings E_t from Fig. 1, but an analogous method[1] can be used to train state embeddings X_t. Following this step, we can use any off-the-shelf model-free RL algorithm to train the internal policy π_i.

[1]For state embeddings, we only use the gradient from the next-state prediction loss

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand.



(a) Chandak et. al. [3] (b) Pritz et. al. [11] (c) Proposed Architecture

Figure 1: Comparison with prior work. The proposed Dual Channel Training (DCT) architecture improves learnt embeddings E_t with a significant effect on the rate and quality of policy learning, as shown in this paper.

We use DDPG [12] for most experiments in this paper. The encoder-decoder model is jointly trained using DCT, with loss gradients flowing through both f and g. The generic loss function is given by,

$$\mathcal{L} = \underbrace{L_1(g(X_t, E_t), S_{t+1})}_{\text{prediction loss}} - \eta \times \underbrace{\frac{1}{N} \times \log P(A_t|f, E_t)}_{\text{reconstruction loss}}, \quad (1)$$

where L_1 is a metric to measure the state prediction loss, N is the number of actions, and $P(A_t|f, E_t)$ is the softmax probability of decoding the embedding E_t to the correct action A_t, as parameterised by f.The multiplier η is a hyperparameter used for trading off the importance between the two loss terms. Complete details can be found in the paper [2]

3 RESULTS: NAVIATION IN 2D MAZE

We present results on a 2-D Maze environment, which an agent with a number of directional actuators is expected to navigate.

3.1 Training Results

Figure 2 presents the training results for 2^{11} actions (11 actuators). We can observe that DCT outperforms all the other baseline algorithms, converging earlier and reaching a higher reward.

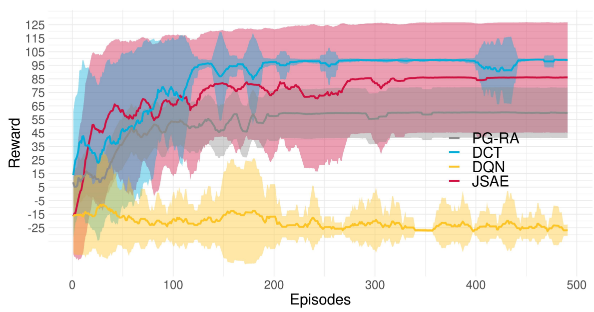


Figure 2: Training results for 2^{11} actions: DCT in blue, JSAE in red, PG-RA in grey, and DQN in yellow. Averages over 10 random seeds.

[2]https://arxiv.org/pdf/2306.15913.pdf

Table 1 presents results over various actions. We can see that even with just 500 episodes (Fig. 2), DDPG over DCT embeddings is able to learn consistently across the actions.

Table 1: Traning results over various actions from 2^6 to 2^{12}.

	2^6		2^{10}		2^{12}	
	μ	σ	μ	σ	μ	σ
DQN	97.38	4.05	-22.03	12.24	-25.92	4.79
PG-RA	97.46	0.46	45.91	65.31	73.15	53.31
JSAE	**98.76**	0.33	82.25	39.05	90.04	27.02
DCT	72.69	54.03	**98.39**	1.09	**99.15**	0.15

4 RESULTS: RECOMMENDER SYSTEMS

We present results from a recommender system task as a second experiment aiming to suggest meaningful products that result in actual purchases for the user.

4.1 Baselines and RL training

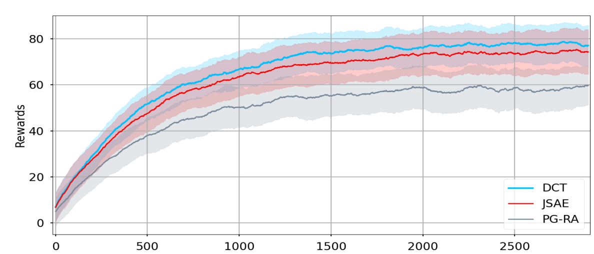


Figure 3: Training curves for the proposed method (DCT) and two baselines, over 5 random seeds.

5 CONCLUSION

Finally, we can conclude from the experiments that DCT is able to learn across a different number of actions consistently. This is validated across 2 diverse environment of navigation and recommender systems. As a part of the investigation, we have also looked at how loss coefficient η affect the structure of embedding.

REFERENCES

[1] Mohammad Mehdi Afsar, Trafford Crump, and Behrouz H. Far. 2021. Reinforcement learning based recommender systems: A survey. *CoRR* abs/2101.06286 (2021). arXiv:2101.06286 https://arxiv.org/abs/2101.06286

[2] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. 2019. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680* (2019).

[3] Yash Chandak, Georgios Theocharous, James Kostas, Scott Jordan, and Philip Thomas. 2019. Learning action representations for reinforcement learning. In *International conference on machine learning*. PMLR, 941–950.

[4] Gabriel Dulac-Arnold, Richard Evans, H. V. Hasselt, Peter Sunehag, Timothy P. Lillicrap, Jonathan J. Hunt, Timothy A. Mann, Théophane Weber, Thomas Degris, and Ben Coppin. 2015. Deep Reinforcement Learning in Large Discrete Action Spaces. *arXiv: Artificial Intelligence* (2015).

[5] Richard Evans and Jim Gao Gao. [n.d.]. DeepMind AI Reduces Google Data Centre Cooling Bill by 40%. ([n. d.]). https://www.deepmind.com/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-by-40

[6] Zhengyao Jiang, Dixing Xu, and Jinjun Liang. 2017. A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint arXiv:1706.10059* (2017).

[7] Jens Kober, J. Andrew Bagnell, and Jan Peters. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* 32, 11 (2013), 1238–1274.

[8] Hongzi Mao, Mohammad Alizadeh, Ishai Menache, and Srikanth Kandula. 2016. Resource management with deep reinforcement learning. (2016), 50.

[9] Hardik Meisheri, Nazneen N Sultana, Mayank Baranwal, Vinita Baniwal, Somjit Nath, Satyam Verma, Balaraman Ravindran, and Harshad Khadilkar. 2022. Scalable multi-product inventory control with lead time constraints using reinforcement learning. *Neural Computing and Applications* 34, 3 (2022), 1735–1757.

[10] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[11] Paul J. Pritz, Liang Ma, and Kin K. Leung. 2020. Joint State-Action Embedding for Efficient Reinforcement Learning. *CoRR* abs/2010.04444 (2020). arXiv:2010.04444 https://arxiv.org/abs/2010.04444

[12] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. 2014. Deterministic policy gradient algorithms. In *International conference on machine learning*. PMLR, 387–395.

[13] Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojtek Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, Timo Ewalds, Dan Horgan, Manuel Kroiss, Ivo Danihelka, John Agapiou, Junhyuk Oh, Valentin Dalibard, David Choi, Laurent Sifre, Yury Sulsky, Sasha Vezhnevets, James Molloy, Trevor Cai, David Budden, Tom Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Toby Pohlen, Dani Yogatama, Julia Cohen, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Chris Apps, Koray Kavukcuoglu, Demis Hassabis, and David Silver. 2019. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/.