ESP-IJACT
ESP International Journal of Advancements in Computational Technology
ISSN: 2583-8628 / Volume 1 Issue 2 September 2023 / Page No: 85-88
Paper Id: IJACT-V1I2P109 / Doi: 10.56472/25838628/IJACT-V1I2P109

Original Article

Comparative Study of Neural Network Architectures in Deep Reinforcement Learning

Ruchi Agarwal
Global Black Belt, Analytics and AI, Microsoft, USA.

Received Date: 28 August 2023 Revised Date: 15 September 2023 Accepted Date: 29 September 2023

Abstract: *This article presents a comprehensive comparative analysis of various neural network architectures employed in deep reinforcement learning (DRL). We examine the efficacy, computational complexity, and scalability of different architectures, including feedforward networks, convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformer-based models. Our study encompasses both value-based and policy-gradient methods, evaluating their performance across a spectrum of environments and tasks. The findings illuminate the strengths and limitations of each architecture, providing insights for researchers and practitioners in the field of DRL.*

Keywords: *Deep Reinforcement Learning, Neural Network Architectures, Feed Forward Neural Networks, Convolutional Neural Networks, Recurrent Neural Networks, Long Short-Term Memory Networks, Attention Mechanisms, Transformers.*

I. INTRODUCTION

Deep Reinforcement Learning (DRL) has emerged as a powerful paradigm for solving complex decision-making problems, achieving remarkable success in various domains such as game playing [1], robotics [2], and autonomous systems [3]. At the core of DRL algorithms lies the neural network architecture, which serves as the function approximator for value functions or policies. The choice of architecture significantly impacts the learning efficiency, generalization capabilities, and overall performance of DRL agents.

This study aims to provide a systematic comparison of different neural network architectures in the context of DRL. We focus on four primary categories of architectures:

- Feedforward Neural Networks (FNNs)
- Convolutional Neural Networks (CNNs)
- Recurrent Neural Networks (RNNs)
- Transformer-based Models

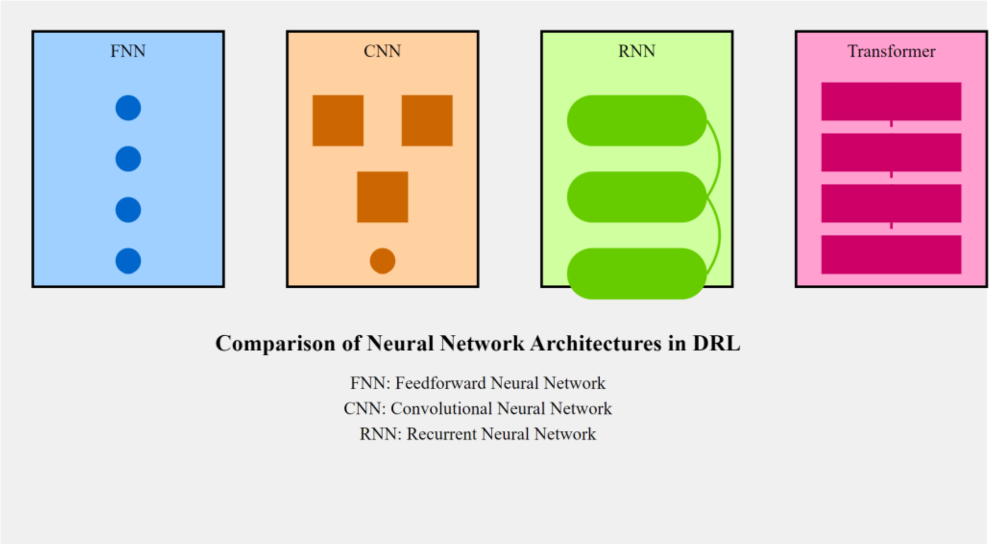

Figure 1: Comparison of Neural Network Architectures in DRL

Each of these architectures possesses unique characteristics that make them suitable for specific types of tasks and environments. Our analysis encompasses both value-based methods, such as Deep Q-Networks (DQN) [4], and policy gradient methods, including Proximal Policy Optimization (PPO) [5] and Soft Actor-Critic (SAC) [6].

II. METHODOLOGY

A. Experimental Setup

We utilize a diverse set of environments from the OpenAI Gym [7] and DeepMind Lab [8] to assess the performance of each architecture across various tasks, including:

- Classic control problems (e.g., CartPole, Acrobot)
- Atari 2600 games
- Continuous control tasks (e.g., MuJoCo environments)
- 3D navigation tasks

For each environment, we implement and train DRL agents using different neural network architectures, maintaining consistency in hyperparameters and training procedures to ensure a fair comparison.

B. Performance Metrics

We evaluate the architectures based on the following metrics:

- Sample efficiency: The number of environment interactions required to achieve a specific performance threshold.
- Asymptotic performance: The maximum performance achieved after a fixed number of training steps.
- Computational complexity: Training time and memory requirements.
- Generalization: Performance on unseen tasks or variations of the training environment.

C. Architecture Specifications

For each category of neural network architecture, we consider the following variants:

Feedforward Neural Networks:

- Multilayer Perceptrons (MLPs) with varying depths and widths
- Dueling networks [9]

D. Convolutional Neural Networks:

- Standard CNNs with different kernel sizes and strides
- Inception-style modules [10]
- ResNet-style architectures [11]
- Recurrent Neural Networks:
- Long Short-Term Memory (LSTM) networks [12]
- Gated Recurrent Units (GRUs) [13]
- Transformer-based Models:
- Standard Transformer architecture [14]
- Gated Transformer-XL [15]

III. RESULTS AND ANALYSIS

A. Value-based Methods

In our experiments with value-based methods, particularly DQN variants, we observed the following trends:

Feedforward Neural Networks:

MLPs demonstrated robust performance across a wide range of tasks, particularly in environments with low-dimensional state spaces. The introduction of dueling architectures consistently improved performance in Atari games, corroborating the findings of Wang et al. [9].

a) Convolutional Neural Networks:

CNNs excelled in tasks with high-dimensional visual inputs, such as Atari games. ResNet-style architectures exhibited superior performance compared to standard CNNs, likely due to their ability to mitigate the vanishing gradient problem in deeper networks.

b) Recurrent Neural Networks:

LSTM and GRU-based architectures demonstrated significant advantages in partially observable environments and tasks requiring memory. However, they often required more samples to converge compared to feedforward architectures.

c) Transformer-based Models:

While relatively new to the DRL domain, transformer-based models showed promising results, particularly in environments with long-term dependencies. The Gated Transformer-XL architecture exhibited superior performance in complex 3D navigation tasks, outperforming RNNs in terms of sample efficiency and asymptotic performance.

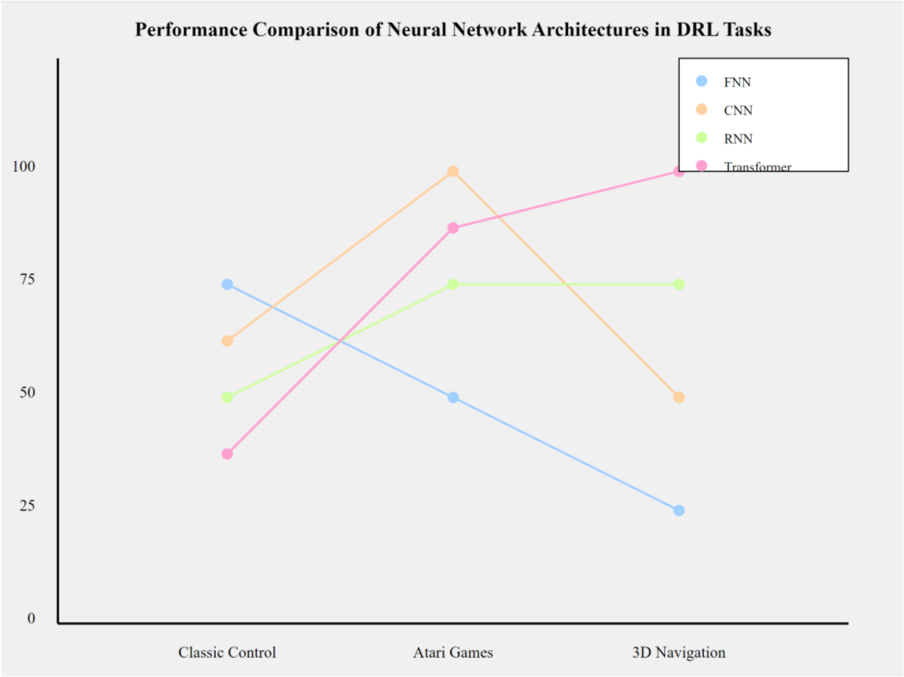

Figure 2: Performance Comparison of Neural Network Architectures in DRL Tasks

B. Policy Gradient Methods

For policy gradient methods, including PPO and SAC, we observed the following:

a) Feedforward Neural Networks:

MLPs remained a strong baseline for continuous control tasks, offering a good balance between performance and computational efficiency.

b) Convolutional Neural Networks:

In visual-based continuous control tasks, CNNs with inception-style modules demonstrated superior sample efficiency compared to standard CNNs.

c) Recurrent Neural Networks:

LSTM-based policies showed significant improvements over feedforward networks in tasks with partial observability, such as certain MuJoCo environments with limited sensor information.

d) Transformer-based Models:

Transformer architectures exhibited strong performance in tasks requiring long-term planning and complex state-action mappings. However, they often incurred higher computational costs compared to other architectures.

C. Computational Complexity

Our analysis revealed a clear trade-off between model complexity and performance. Transformer-based models, while often achieving superior results, required significantly more computational resources and training time compared to simpler architectures. CNNs and RNNs occupied a middle ground, offering a balance between performance and efficiency.

D. Generalization

In terms of generalization to unseen tasks or environment variations:

- CNNs demonstrated strong transfer learning capabilities in visual domains.
- Transformer-based models showed promising results in generalizing to variations in 3D navigation tasks.
- RNNs exhibited robust performance when transferring to environments with different temporal dynamics.

IV. DISCUSSION

Our comprehensive study reveals that the optimal choice of neural network architecture for DRL is highly dependent on the specific characteristics of the task and environment. Key findings include:

- The enduring effectiveness of relatively simple architectures, such as MLPs, in many DRL scenarios, challenging the notion that more complex models are always beneficial.
- The superior performance of CNNs in visual-based tasks, with ResNet-style architectures offering particular advantages in deep networks.
- The critical role of recurrent architectures in partially observable environments and tasks requiring memory.
- The emerging potential of transformer-based models, particularly in scenarios involving long-term dependencies and complex state-action mappings.

These results underscore the importance of carefully considering the trade-offs between model complexity, computational requirements, and task-specific performance when selecting neural network architecture for DRL applications.

V. CONCLUSION AND FUTURE WORK

This study provides a comprehensive comparison of neural network architectures in the context of deep reinforcement learning. Our findings offer valuable insights for researchers and practitioners in selecting appropriate architectures for specific DRL tasks.

Future work should explore the potential of hybrid architectures that combine the strengths of different model types. Additionally, investigating the impact of advanced training techniques, such as meta-learning and multi-task learning, on the performance of various architectures presents an exciting avenue for further research.

VI. REFERENCES

[1] Mnih, V., et al. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533.
[2] Levine, S., et al. (2016). End-to-end training of deep visuomotor policies. The Journal of Machine Learning Research, 17(1), 1334-1373.
[3] Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. The International Journal of Robotics Research, 32(11), 1238-1274.
[4] Mnih, V., et al. (2013). Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.
[5] Schulman, J., et al. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.
[6] Haarnoja, T., et al. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. arXiv preprint arXiv:1801.01290.
[7] Brockman, G., et al. (2016). OpenAI Gym. arXiv preprint arXiv:1606.01540.
[8] Beattie, C., et al. (2016). DeepMind Lab. arXiv preprint arXiv:1612.03801.
[9] Wang, Z., et al. (2016). Dueling network architectures for deep reinforcement learning. In International conference on machine learning (pp. 1995-2003). PMLR.
[10] Szegedy, C., et al. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).
[11] He, K., et al. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
[12] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
[13] Cho, K., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.
[14] Vaswani, A., et al. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
[15] Parisotto, E., et al. (2020). Stabilizing transformers for reinforcement learning. In International Conference on Machine Learning (pp. 7487-7498). PMLR.