RADemics

# Combining Neural Architecture Search and Reinforcement Learning for Optimized Autonomous System Performance

Sindhusaranya Balraj, S. Priyadharshini
SONA COLLEGE OF TECHNOLOGY, VELALAR COLLEGE OF ENGINEERING AND TECHNOLOGY

# 12. Combining Neural Architecture Search and Reinforcement Learning for Optimized Autonomous System Performance

[1]Sindhusaranya Balraj, Assistant Professor, Department of Computer Science and Engineering, Sona College of Technology, Salem, Tamil Nadu - 636006, India, sindhusaranyabalraj@gmail.com

[2]S. Priyadharshini, Assistant Professor, Department of Computer Science and Engineering, Velalar College of Engineering and Technology, Thindal, Erode, Tamil Nadu, India,

## Abstract

This book chapter explores the integration of Neural Architecture Search (NAS) and Reinforcement Learning (RL) to optimize the performance of autonomous systems. By combining the adaptive capabilities of NAS with the decision-making prowess of RL, this approach aims to create more efficient, scalable, and intelligent systems capable of tackling complex tasks. The chapter delves into the synergies between NAS and RL, highlighting their potential to improve system robustness, adaptability, and generalization across dynamic environments. It addresses the challenges involved in integrating these technologies, such as computational complexity, overfitting, and the balance between exploration and exploitation. Through comparative analyses with traditional methods, the chapter underscores the advantages of NAS-RL systems in benchmark tasks, showcasing their superiority in task-specific optimization and dynamic task adaptation. This comprehensive discussion provides valuable insights for researchers and practitioners aiming to advance autonomous system design through cutting-edge AI techniques.

**Keywords:**

Neural Architecture Search, Reinforcement Learning, Autonomous Systems, System Optimization, Robustness, Benchmark Tasks

## Introduction

Autonomous systems have become integral to a wide range of applications, including robotics, self-driving cars, drones, and intelligent decision-making systems [1,2]. These systems are designed to operate independently, adapting to their environment and executing complex tasks without human intervention [3-5]. The rapid advancement in artificial intelligence (AI) and machine learning (ML) has enabled the development of such systems, yet the challenges of optimizing their performance in dynamic and unpredictable environments persist [6,7]. As autonomous systems tackle increasingly complex real-world tasks, ensuring their efficiency, robustness, and scalability was critical [8]. One promising approach to achieving these objectives was by combining Neural Architecture Search (NAS) with Reinforcement Learning (RL) [9]. This combination aims to enhance the decision-making and adaptability of autonomous systems while optimizing their underlying neural architectures [10,11].

Neural Architecture Search (NAS) has emerged as a transformative technique that automates the design of neural networks [12]. Traditionally, neural network architectures were manually designed, relying on the expertise of researchers and practitioners [13]. However, this manual process can be time-consuming and limited in its ability to explore diverse design spaces [14]. NAS automates the discovery of optimal architectures by leveraging search algorithms, allowing for the exploration of a wide range of neural network configurations [15,16]. By applying NAS, researchers can identify architectures that are well-suited to specific tasks, leading to improved performance and reduced resource consumption [17,18]. NAS allows for the development of highly specialized neural architectures that can adapt to the unique demands of different environments, thus improving the efficiency and effectiveness of these systems [19,20].

Reinforcement Learning (RL) was a branch of machine learning that focuses on training agents to make decisions by interacting with an environment [21,22]. Through trial and error, RL agents learn to take actions that maximize cumulative rewards [23]. This makes RL particularly suitable for autonomous systems that must continuously adapt to changing conditions and optimize their decision-making over time [24]. RL has been widely used in various domains, such as robotics, game playing, and natural language processing, to enable autonomous agents to learn from experience. However, the integration of RL into autonomous systems presents its own set of challenges, such as balancing exploration and exploitation, mitigating overfitting, and ensuring long-term stability [25].