# Neural Architecture Search and tinyML

Daniel Duclos-Cavalcanti

***Abstract*—There is no denying the increasing success that Deep Neural Networks (DNNs) have displayed across various tasks such as image classification, speech recognition, machine translation and many others. This progress can be attributed largely due to *architecture engineering*, which is a process that requires immense domain expertise, intuition and some amount of trial and error. This is not ideal as it is both time consuming and error-prone. Neural Architecture Search (NAS) rises as the next logical step aiming to both automate architecture discovery and further the understanding of the inner workings of DNNs. This field has grown remarkably within the last 5 years and among the many challenges of NAS, there are concerns regarding computational costs and time feasability. This however becomes a different situation within the context of *tinyML*, which is an expanding field at the intersection of machine learning and embedded systems. Due to the resource constrained conditions involved in most embedded devices and tinyML applications, the stages concerning training and performance evaluation are substantially faster in comparison to the more complex models used in the research of NAS. Thus presenting an interesting opportunity to explore NAS applications within this different paradigm. Throughout this work, an overview of existing research in NAS, specifically concerning evolutionary algorithms and reinforcement learning will be presented, as well as highlight relevant applicabilities to tinyML.**

## I. Introduction

The progress of deep learning has brought a continuous demand for architecture engineering. Given the growing complexity of designs, this process has become increasingly challenging and more susceptible to errors. However, after the work in [1] proposed by Google, it is shown for the first time that NAS algorithms have the potential to find models that rival the current state of the art. Since then, many different methods and approaches to have appeared.

To better visualize NAS, one can categorize it within three dimensions [2]:

- **Search Space**: defines all the possible architectures that can in principle be considered.
- **Search Strategy**: defines how the *search space* is explored by the algorithm.
- **Performance Estimation Strategy**: defines how performance is evaluated at every architecture iteration.

### A. Search Space

NAS is an optimization problem, whose *search space* is the defining factor to the difficulty of its solution. The smaller the *search space* is, the faster the search algorithm may converge into a result, as well as requiring less computational resources. However, this also comes at the cost of less freedom to explore unseen architectures and also may limit the complexity of the design.

The simplest approach is to define an architecture as a *chain-structured neural network*, which essentially consists of a sequence of layers whose inputs are the output of their preceding layer. In this case the space is parametrized by maximum number of layers, type of operations per layer, and hyperparameters conditioned on the chosen operation.

There have been approaches that

## II. State of the Art

In this section an analysis of the available literature on the topic is done. This section may be split or subdivided into several sections or subsections.

### A. Subsection Heading Here

Subsection text here.

*1) Subsubsection Heading Here:* Subsubsection text here.

## III. Conclusion

Put the conclusions of the work here. The conclusion is like the abstract with an additional discussion of open points.

## REFERENCES

[1] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.
[2] M. Wistuba, A. Rawat, and T. Pedapati, "A survey on neural architecture search," *arXiv preprint arXiv:1905.01392*, 2019.