Optimisation performance – accuracy / recall.
Optimisation time – CPU / GPU time.
Model complexity.

Literature Review

Christopher MacKinnon

November 2020

1 Hyper-Parameters

Why noisy?

Hyper-parameter optimisation is traditionally considered a noisy, black-box optimisation problem. This means the problem is considered only in terms of inputs and outputs without consideration for the inner operation. Although the inner operation of Machine Learning (ML) systems can be observed, it is generally considered too complex and therefore functionally equivalent to a black-box. However, some more recent approaches have attempted to move away from this perspective [28], [36].

Need intro in this para.
↓
What are hyper-parameters

Non sequitur
↓
explainability does not translate to optimal params

A manual search combing a grid search and expert knowledge has been the standard approach to this problem in the past, with model designers relying on experience and intuition to guide the optimisation process. It was shown that a random search could perform as well as, or better than, grid search in the same time [7] and this type of search has become a minimum benchmark or starting point for many hyper-parameter optimisation techniques [23][11]. Recently, taking advantage of the increase in parallel and cloud computing power has become a key aspect of these systems, requiring robust methods that can operate efficiently at scale, with the SOTA in some application requiring thousands of GPU days of computation [26], [28], [38], [41] . Hyper-parameters can contain continuous, discrete and categorical variables requiring flexible approaches to find optimal solutions across a wide range of architectures and algorithms.

What is this?

There has been an increase in interest in Neural Architecture Search (NAS) as these methods began to outperform human designed models. [41][38] These systems are able to design at scale creating intricate topologies for large, deep networks, while discovering novel variants in the components used to build these networks. [25] [32] In the context of computer vision this problem has been investigated significantly over the last four years, with a range approaches from genetic algorithms to reinforcement learning being applied to the problem. NAS problems are often constrain the search space to a set of cells stacked in a predefined structure. These structures are based upon successful past implementation in these domains [15], [27], [41], [42]. The design of the search space over which a method implemented can have a very significant impact on the success of a search procedure and is still an important aspect of the NAS process.

Don't start a sentence with citations