Chapter 9

Deep Learning Techniques and Optimization Strategies in Big Data Analytics:
Automated Transfer Learning of Convolutional Neural Networks Using Enas Algorithm

Murugan Krishnamoorthy
https://orcid.org/0000-0002-6751-380X
Anna University, India

Bazeer Ahamed B.
https://orcid.org/0000-0003-1559-8386
Balaji Institute of Technology and Science, India

Sailakshmi Suresh
Anna University, India

Solaiappan Alagappan
Anna University, India

ABSTRACT

Construction of a neural network is the cardinal step to any machine learning algorithm. It requires profound knowledge for the developer in assigning the weights and biases to construct it. And the construction should be done for multiple epochs to obtain an optimal neural network. This makes it cumbersome for an inexperienced machine learning aspirant to develop it with ease. So, an automated neural network construction would be of great use and provide the developer with incredible speed to program and run the machine learning algorithm. This is a crucial assist from the developer’s perspective. The developer can now focus only on the logical portion of the algorithm and hence increase productivity. The use of Enas algorithm aids in performing the automated transfer learning to construct the complete neural network from the given sample data. This algorithm proliferates on the incoming data. Hence, it is very important to inculcate it with the existing machine learning algorithms.

DOI: 10.4018/978-1-7998-1192-3.ch009
INTRODUCTION

Artificial Intelligence (AI), a hot topic among the computer society these days. This revolutionary domain indirectly depends on Deep Learning to formulate the structure of the artificial neural network, the brain. Deep learning is a process and a way to automate Predictive Analytics. Traditional machine learning algorithms are linear and are stacked in a hierarchy of increased abstraction and complexity. AutoML alleviates human exertion by computerizing the layout of ML algorithms (Afridi, Ross, and Shapiro, 2018). It is prevalent for the implementation of deep learning framework. But this perception incurs exalted computational cost. To reconcile this issue, we resolve to Transfer neural techniques that employs comprehension from preceding tasks to accelerate design of neural networks. We broaden probe routines to enable simultaneous practice over various tasks and then transit the probe strategy to new tasks. This proposal deduces the coupling time of a task training by significant degrees on many tasks.

The system controller is modelled over generic engineering composition and parameters, along with task-related decisions encrypted in the task implantation (Ahamed and Ramkumar, 2016). For any given task, the parameters of the corresponding task controller are reloaded and new implantations are randomly activated for the new task. The parameters of the controller are synchronized with the new task implantations. By implantation for new tasks, the application that biases towards process actions that excelled well on other comparable tasks is learnt by the controller. This is the performance of transfer controller.

Designing these deep learning models is a difficult task on its own and it is not too farfetched, that it is a process of trial and error, as there is no fixed methodology to design a model that is guaranteed to perform well. Thus the focus of this implementation is to provide a modular and extensible framework that can automatically design high performing deep learning models.

BACKGROUND STUDY

Neural Networks and Domain Specific Language

Neural networks a set of logic and knowledge based approach sequenced, modelled based on the brain of human (Ahamed and Yuvaraj, 2018). They recognize by interpreting patterns, sensory data through machine perception. Commercial applications technologies generally focused on solving complex pattern or signal processing problems such as speech, handwriting, oil exploration data analysis, facial recognition and weather predictions. A Domain Specific Language (DSL), computer programing restricted expressiveness focused on a specific domain. DSLs provides significant gain in Application productivity, creativity developers, portability and performance. DSLs offer pre-defined abstractions represent concepts from application domain. A programmer uses one or more of the DSLs write the programs using specific domain constructs and notations. In additional ability to use domain knowledge to apply static & dynamic optimizations to a program (Cessac et. al, 2016). DSL specifically targeted, machine learning, abstractions to define neural networks, capture high level information used to increased productivity, performance and expose parallelism. It moves the programmers approach from a low level detailing that are not the focus of the system in development, lets them target on work the solution for the problem at hand.