John Ziegler

07/29/2022

CSCI 6600 – Selected Topics/Cloud/Big Data

Comprehension Assessment 4

MLlib: Machine Learning in Apache Spark

Introduction

Datasets in the coming of age have grown largely and become more complex and techniques using the MapReduce function have been proven to be effective with machine learning as well, Spark being one of these. Spark is a fault-tolerant cluster computer system that has built in API’s for Java, Scala, Python, and R. Spark is seemed to be ideal for large-scale machine learning. The MLlib, Spark’s distributed machine learning library, and the largest library with his application. It basis it’s efficiency off data-parallelism or model-parallelism to operate and store on data or models. It also has built in, and scalable applications for classifications, regressions, collaborative filtering, clustering, and dimensionality reduction. Spark is designed for ideally iterative computation, and allows the implementations of large scale, efficient machine learning algorithms, since they are usually iterative.

History and Growth

Spark was invented at UC Berkeley AMPLab and open-sourced in 2010. Despite being efficient for iterative computation, it lacked a suite of scalable learning algorithms until the invention of the MLlib. MLlib has been packaged with Spark and was also developed at UC Berkely by 11 different contributors. MLlib now has 140 contributors, and truly demonstrates how the application can grow with time with an open-source community

Core Features

Supported Methods and Utilities, provided by MLlib, gives fast, distributed implementations of learning algorithms including linear models, naïve Bayes, and ensembles of decision trees for classification and regression problems. Alternating least squares with explicit and implicit feedback for filtering. It also provides low-level primitives and basic utilities for convex optimizations, distributed linear algebra, statistical analysis, and feature extraction. In addition to learning algorithms, we also have Algorithmic Optimizations, which supports distributed learning and prediction processes. For example, the ALS algorithm makes use of blocking to reduce Java Virtual Machine garbage collection overhead and leverages higher-level linear algebra operations. The Pipeline API involve a sequence of data preprocessing, feature extraction, model fitting, and validation stages. The Spark API was designed so that when dealing with large datasets, we didn’t have to cobble together the end-to-end pipeline, which ultimately is labor intensive and expensive in terms of network overhead. Spark Integration benefits from the components of spark at the lowest level. It provides a general execution engine with more then 80 operations for transforming data. It also leverages high-level libraries packaged with Spark such as SQL for data integration functionality.