COMP 543, Tools and Models for Data Science

Chengyin Liu, cl93

Research #6, TensorFlow

This paper [1] describes the interface and development process of TensorFlow, which is a general-purpose numerical computation library open-sourced by Google in 2015. TensorFlow has a flexible architecture as an interface for expressing machine learning algorithms, and an implementation for executing such algorithms. Single-device, multi-device, and distributed executions are explained in this paper. And it also presents how to make extensions and do optimizations based on TensorFlow, which enables users to focus on building the machine learning models with little efforts on heterogeneous platforms.

In 2016, Yuan Tang developed and presented TF.Learn [2], a high-level Python module for distributed machine learning inside TensorFlow, because many users may find it hard to start building deep learning models using original TensorFlow. An easy-to-use Scikit-learn [3] style interface is provided in his paper to simplify the process of deploying and evaluating machine learning models.

Examples of TF.Learn are presented in this paper, which shows that it utilizes Python's object-oriented characteristics. TF.Learn also integrates state-of-art machine learning algorithms built on TensorFlow, for both supervised and unsupervised problems. As a result, TF.Learn brings a more friendly way of implementing machine learning algorithms to non-specialists and researchers in a structured environment.

References:

[1] Abadi, Martín, et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." arXiv preprint arXiv:1603.04467 (2016).

[2] Tang, Yuan. "TF. Learn: TensorFlow's High-level Module for Distributed Machine Learning." arXiv preprint arXiv:1612.04251 (2016).

[3] Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." Journal of machine learning research 12.Oct (2011): 2825-2830.