High-Performance Machine Learning: Systems and Applications

CSc G0815, 3 Credits, Spring 2018, Tuesdays 4:50-7:20

Professor Jianting Zhang (jiazhang@ccny.cuny.edu, 212-650-6175)

Increasingly available Big Data, massively parallel computing power, new deep learning algorithms have dramatically changed practices of Machine Learning (ML) in the past few years. While traditional ML algorithms such as Random Forests (RF) and Support Vector Machines (SVMs) have been applied to large-scale datasets in an efficient and scalable way, Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) in Deep Learning (DL) frameworks have reached unprecedented accuracies in vision, speech and language data processing and are quickly applied to many other applications. Subsequently, open source machine learning systems, such as Google TensorFlow, Microsoft CNTK, Apache MXNet and Caffe/Caffe2, have becoming popular in both research and industry. These systems rely on multi-level parallel and distributed computing power on modern hardware architectures to provide desired high performance in training and testing, which are very different from traditional ML systems that execute serial programs on single CPUs. These have imposed significantly technical challenges on ML learners and practitioners.

This course introduces ML from a system-oriented aspect. After a brief review of selected ML algorithms (including both traditional and DL-based), the course will provide an in-depth analysis of Google TensorFlow, including both architecture and components, as an example of thorough understanding of modern high-performance ML systems. Students will further learn how ML algorithms can be efficiently implemented in TensorFlow in a scalable manner by utilizing TensorFlow's infrastructure and understand various design tradeoffs between system abstractions, extensibility and performance. The next topic focuses on utilizing TensorFlow's extensible architecture to develop custom operations on multi-core CPUs and GPUs that can be utilized interactively in Python though demonstrative examples. Selected ML applications on TensorFlow will be presented in concurrent with term project during the last 1/3 of the semester.

For grading, the largest chunk is the group-based term project (40%), which can either extend TensorFlow's capability by developing carefully selected operations or build end-to-end machine learning applications using either TensorFlow or other ML systems of team's choice. There will be two individual or group projects (30%) to help students practice TensorFlow skills and get ready for the term project. Program templates will be provided to ease the learning process. A mid-exam (20%) is planned to test the basic knowledge of the covered topics during the first half of the course. The remaining 10% is class participation.

The tentative topics and schedule are the following:

- Introduction to high-performance ML: Big Data, Parallel Computing, ML algorithms, ML systems
- Review of ML algorithms: Decision Tree (DT) and Random Forests (RF), Support Vector Machines (SVM), CNN, RNN (Gated Recurrent Unit (GRU) and/or Long-Short Term Memory (LSTM)).
- TensorFlow: Architecture and Components
- ML algorithms on TensorFlow: implementation and performance with different hardware configurations
- Extending TensorFlow with custom operators and Python integration
- Case studies of high-performance ML applications (concurrent with student term projects)

Textbook & References

- No textbooks are required but references will be provided (with online PDF/audio/video resources)
- Slides and links to research articles will be provided before classes (through Blackboard system).
- Reference 1: Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep Learning. ISBN 978-0262035613 (http://www.deeplearningbook.org/)
- Reference 2: Aurellien Geron, Hands-on Machine Learning with Scikit-Learn & TensorFlow. ISBN 978-1491962299 (http://shop.oreilly.com/product/0636920052289.do)

Prerequisites and Logistics:

- *Reasonable proficiency in programming* and *enthusiasm in machine learning and high performance computing* are essential. Previous experiences with **C/C++** and Python are **preferred** but **not required**.
- *Examples and templates* for individual/group projects will be provided during the first half of the class to help students get familiar with TensorFlow programming (including Python, C/C++ and CUDA code).
- Remote accesses to multi-core CPU and GPU hardware and software will be provided by the instructor.