Technical Perspective: Scaling Machine Learning via Compressed Linear Algebra

Zachary G. Ives
University of Pennsylvania
zives@cis.upenn.edu

Demand for more powerful "big data analytics" solutions has spurred a great deal of interest in the core programming models, abstractions, and platforms for next-generation systems. For these problems, a complete solution would address data wrangling and processing, and support analytics over data of any modality or scale. It would support a wide array of machine learning algorithms, but also provide primitives for building new ones. It should be customizable, scale to vast volumes of data, and map to modern multicore, GPU, co-processor, and compute cluster hardware.

In pursuit of these goals, novel techniques and solutions are being developed by machine learning researchers (e.g., high-performance libraries like Theano [6], runtime systems like GraphLab [5]), in the database and distributed systems research communities (e.g., distributed data analytics engines like Spark [7] and Flink [3]), and in industry by major technology players (e.g., Google's TensorFlow [1] and IBM/Apache's SystemML [4]). These libraries and platforms support multiple development languages, provide abstract datatypes for machine learning over data, and include compilers and runtime systems optimized for distributed execution on modern hardware.

The database community excels in developing techniques for cost-estimating and optimizing declarative programs, and in exploiting *data independence* to optimize data placement and layout for performance. Elgohary et al's work on "Scaling Machine Learning via Compressed Linear Algebra," which appeared in the Proceedings of the VLDB Endowment [2], was conducted within IBM and Apache's SystemML declarative machine learning project. It shows just how effective such database techniques can be in a machine learning setting. The authors observe that the core data objects in machine learning – feature matrices, weight vectors – tend to have repeated values as well as regular structure, and may be quite large. Machine learning tasks over such data are composed from lower-level linear algebra operations. Such operations generally involve repeated floating-point computation that today are *bandwidth-limited*, by the ability of the CPU to traverse large matrices in RAM.

The authors' solution is to develop a compressed representation for matrices, as well as compressed linear algebra operations that work directly over the compressed matrix data. Together, these reduce the bandwidth required to perform the same computations, thus speeding performance dramatically. The paper cleverly adapts ideas first developed in relational database systems — column-oriented compression, sampling-based cost estimation, trading between compression speed and compression rate — to build an elegant implementation.

The paper makes a number of key contributions. First, the authors identify a set of linear algebra primitives shared by multiple distributed machine learning platforms and algorithms. Second, they develop compression techniques not only for single columns in a matrix, but also "column grouping" techniques that capitalize on correlations between columns. They show that offset lists and run-length encoding offer a good set of trade-offs between efficiency and performance. Third, the paper develops cache-conscious algorithms for matrix multiplication and other operations. Finally, the paper shows how to estimate the sizes of compressed matrices and to choose an effective compression strategy. Together, these techniques illustrate how database systems concepts can be adapted to great benefit in the machine learning space.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. A. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: A system for large-scale machine learning. In *OSDI*, pages 265–283, 2016.

[2] A. Elgohary, M. Boehm, P. J. Haas, F. R. Reiss, and B. Reinwald. Compressed linear algebra for large-scale machine learning. *PVLDB*, 9(12):960–971, 2016.

[3] S. Ewen, K. Tzoumas, M. Kaufmann, and V. Markl. Spinning fast iterative data flows. *Proc. VLDB Endow.*, 5(11):1268–1279, 2012.

[4] A. Ghoting, R. Krishnamurthy, E. P. D. Pednault, B. Reinwald, V. Sindhwani, S. Tatikonda, Y. Tian, and S. Vaithyanathan. Systemml: Declarative machine learning on mapreduce. In *ICDE*, pages 231–242, 2011.

[5] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J. M. Hellerstein. Distributed GraphLab: A Framework for Machine Learning and Data Mining in the Cloud. *PVLDB*, 2012.

[6] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[7] M. Zaharia, M. Chodhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: Cluster computing with working sets. In *HotCloud*, 2010.