fib-tf: A TensorFlow-based Cardiac Electrophysiology Simulator

Shahriar Iravanian

1 Emory University

Summary

fib_tf is a Python package developed on top of the machine-learning library TensorFlow for cardiac electrophysiology simulation (Abadi et al. 2015). While TensorFlow is primarily designed for machine learning, it also provides a framework to perform general-purpose multidimensional tensor manipulation.

The primary goal of fib_tf is to test and assess the suitability of TensorFlow for solving systems of stiff ordinary differential equations (ODE), such as those encountered in cardiac modeling. It mainly targets massively parallel hardware architectures (e.g., Graphics Processing Units).

fib_tf solves the monodomain reaction-diffusion equations governing cardiac electrical activity by a combination of the finite-difference and explicit Euler methods. It is used to simulate two cardiac ionic models: the 4-variable Cherry-Ehrlich-Nattel-Fenton canine left-atrial and the 8-variable Beeler-Reuter ventricular models (Cherry et al. 2007; Beeler and Reuter 1977).

fib_tf serves as a testbed to try various general and TensorFlow-specific optimization techniques. We showed that enabling Just-In-Time (JIT) compilation significantly improves the performance. Moreover, by applying a multitude of optimization methods, including dataflow graph unrolling, the Rush-Larsen method (Rush and Larsen 1978), the Chebyshev polynomials approximation, and multi-rate integration, we have achieved a performance within a factor 2-3 of hand-optimized CUDA codes. The motivation behind and the details of each method are described in the documentation.

Based on our experiments, TensorFlow applicability is not limited to the machine-learning domain. TensorFlow is a valuable tool for the development of efficient and complex ODE solvers. fib_tf can act as a framework and model for such solvers. Especially, it is useful in rapid prototyping and testing of new algorithms and methods.

Acknowledgements

References

Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, et al. 2015. “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.” https://www.tensorflow.org/.

Beeler, G W, and H Reuter. 1977. “Reconstruction of the action potential of ventricular myocardial fibres.” The Journal of Physiology 268 (1):177–210. https://doi.org/10.1113/jphysiol.1977.sp011853.
Cherry, Elizabeth M., Joachim R. Ehrlich, Stanley Nattel, and Flavio H. Fenton. 2007. “Pulmonary vein reentry—Properties and size matter: Insights from a computational analysis.” Heart Rhythm 4 (12):1553–62. https://doi.org/10.1016/j.hrthm.2007.08.017.

Rush, Stanley, and Hugh Larsen. 1978. “A Practical Algorithm for Solving Dynamic Membrane Equations.” IEEE Transactions on Biomedical Engineering BME-25 (4):389–92. https://doi.org/10.1109/TBME.1978.326270.

Iravanian, (2018). fib-tf: A TensorFlow-based Cardiac Electrophysiology Simulator. Journal of Open Source Software, 3(26), 719. https://doi.org/10.21105/joss.00719
Iravanian, S.: fib-tf: a TensorFlow-based cardiac electrophysiology simulator. J. Open Source Softw. 3(26), 719 (2018) CrossRefGoogle Scholar. Teaching cardiac electrophysiology modeling to undergraduate students: laboratory exercises and GPU programming for the study of arrhythmias and spiral wave dynamics. Adv. Physiol. [REVIEW]: fib-tf: A TensorFlow-based Cardiac Electrophysiology Simulator #719. Closed. 36 tasks done. The initial reason to develop fib_tf was as a test bed for new ideas on cardiac electrophysiology simulation, like adding a new ionic current to a cardiac model. This is actually how we use it in our academic work (no publication yet, but we are working on a paper on enhancing the Courtemanche atrial model). It is much easier to test ideas in a scripting language like python with all its scientific computing ecosystem than in C++. However, fib_tf can also be useful in assessing the feasibility of writing ODE solvers in TensorFlow. The main power of TensorFlow is its ability to run on multiple GPUs or even on a distributed system with multiple CPUs, each with one or more GPUs. TensorFlow Lite is the lightweight version which is specifically designed for the mobile platform and embedded devices. TFLite models are much faster, smaller in size, and less computationally expensive. We cannot deploy heavy weighted models on ... Based on your years of experience and your career goal, you can select the courses. To know more about whether your profile. (Continue reading)... tf.distribute.Strategy is a TensorFlow API to distribute training across multiple GPUs, multiple machines or TPUs. Using this API, you can distribute your existing models and training code with minimal code changes. tf.distribute.Strategy has been designed with these key goals in mind. You can use tf.distribute.Strategy with very few changes to your code, because the underlying components of TensorFlow have been changed to become strategy-aware. This includes variables, layers, models, optimizers, metrics, summaries, and checkpoints. CommunicationImplementation.RING is RPC-based and supports both CPUs and GPUs. CommunicationImplementation.NCCL uses NCCL and provides state-of-art performance on GPUs but it doesn't support CPUs.