Geoffrey Fox: ML for CI and CI for ML

Feb 20 2020

Award #: 1443054

- **ML for CI** (I used **MLforHPC**) will have transformative impact on computational science improving molecular dynamics simulation performance by large factor up to 2. 10^9 reported https://arxiv.org/pdf/2001.08055.pdf
  - Part of **Systems for ML and ML for Systems** in Dean's talk at NeurIPS 2017 for Big Data applications
  - I survey my collaborations including a classification of much current work and roadmap
  - Show factors of 10^{4-5} for short time (recurrent) and long time (fully connected networks)
- **CI for ML** (I used **HPCforML**) is well studied and successful as Machine Learning has many similarities with simulations that can be exploited
  - Need to integrate Big Data technology (from Apache Software Stack to Containers) with simulations

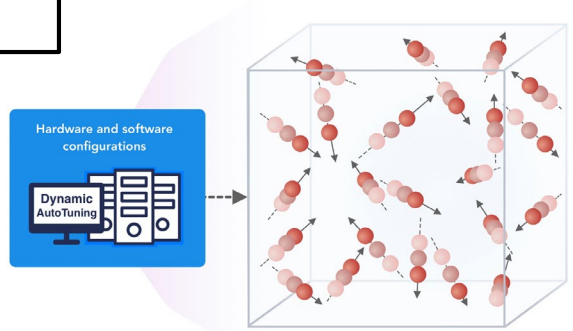
Geoffrey Fox: Status of ML for CI and CI for ML

Award #: 1443054

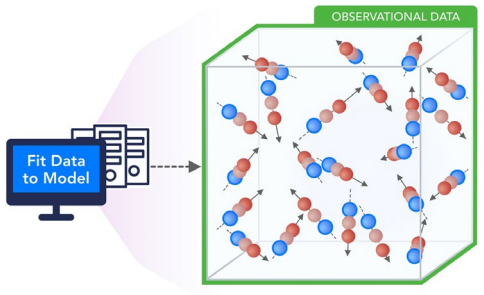
- **What are the challenges for ML for CI and CI for ML?**
  - Increasing importance of deep learning and dominance of Industry in deep learning algorithms and systems needs to be understood
  - Implications of new AI accelerators
- **What are the challenges and opportunities to form cross-disciplinary collaborations for ML for CI and CI for ML?**
  - ML for CI: Speedups of up to 2. 10^9 reported but need significant changes in application code while nature and requirements of cyberinfrastructure unclear.
  - Thus **must have interdisciplinary teams** to both develop new approaches to applications and give requirements to Cyberinfrastructure
- **How do you see the future of ML for CI and CI for ML?**
  - Both will have transformative impact on computational and data science with effective performance of zetta and yottascale expected from ML for CI.
  - Dynamic deep integration of simulation and AI will need nontrivial new middleware
  - **Look into MLPerf**: Currently I think that DoE and I represent CI research community

1. Improving Simulation with Configurations and Integration of Data

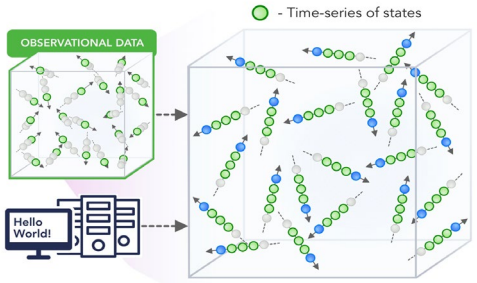
1.1 MLAutotuningHPC – Learn configurations



1.2 MLAutotuningHPC – Learn models from data

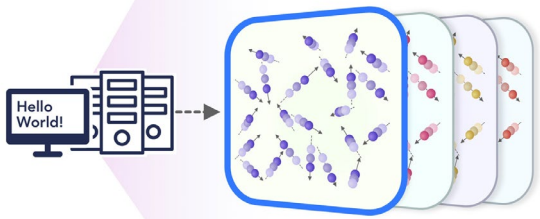


1.3 MLaroundHPC: Learning Model Details (ML based data assimilation)

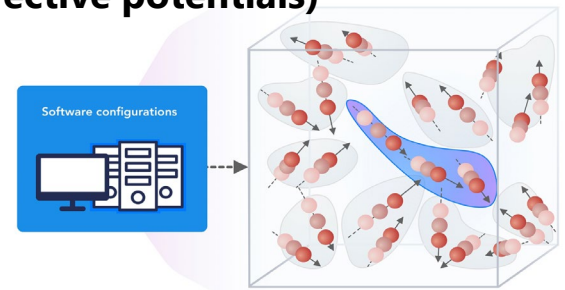


2. Learn Structure, Theory and Model for Simulation

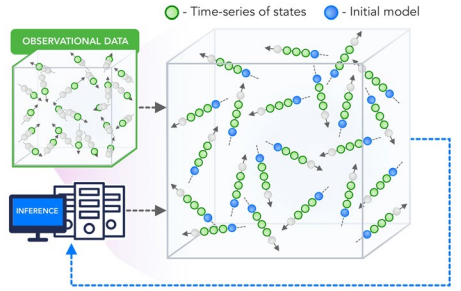
2.1 MLAutotuningHPC – Smart ensembles



2.2 MLaroundHPC: Learning Model Details (coarse graining, effective potentials)



2.3 MLaroundHPC: Improve Model or Theory

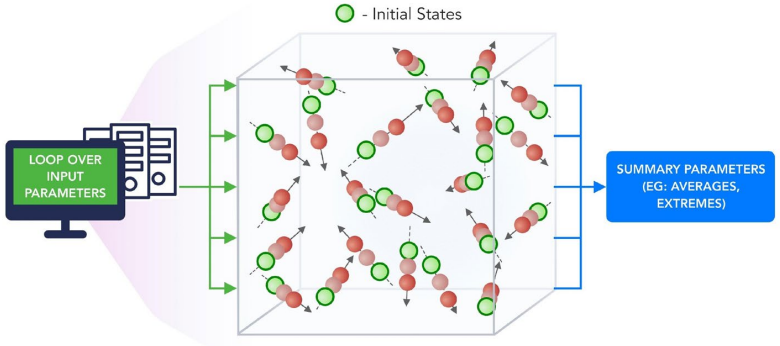


3. Learn Surrogates for Simulation

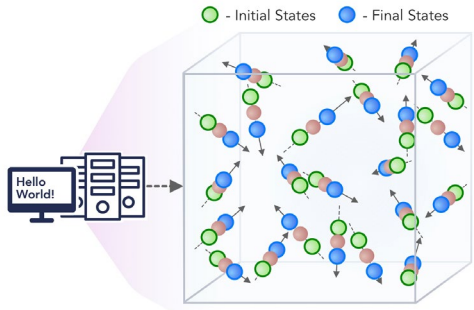
Work with Shantenu Jha
https://arxiv.org/abs/1909.13340
https://arxiv.org/abs/1909.02363

100 refs classified in 3 high level and 8 detailed areas

3.1 MLaroundHPC: Learning Outputs from Inputs (parameters)



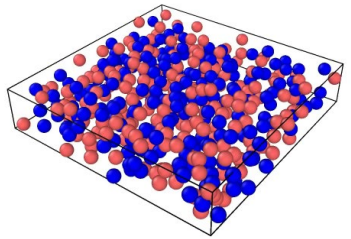
3.2 MLaroundHPC: Learning Outputs from Inputs (fields)



INPUT

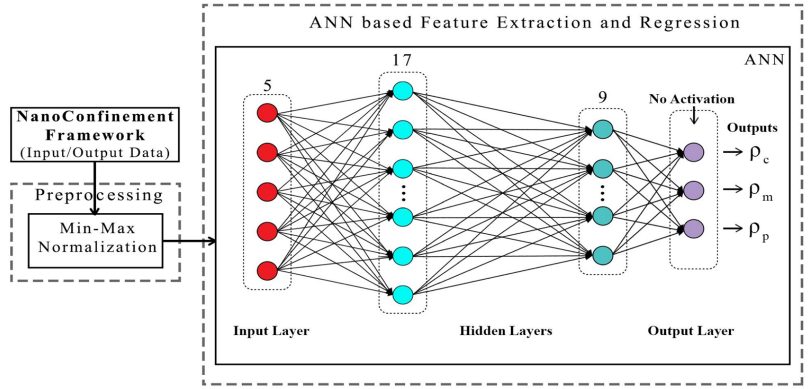
OUTPUT

Examples of ML for CI (work with JCS Kadupitiya, Vikram Jadhao)



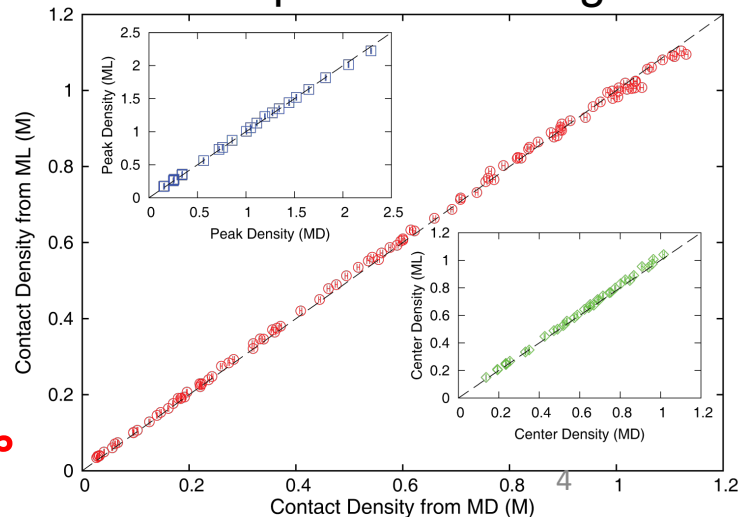
- Extraction of **ionic structure in electrolyte solutions confined by planar and spherical surfaces**.
- Classic HPC code written with C++ and accelerated with hybrid MPI-OpenMP.

The Learning Net



- Uses quite small Multi-layer perceptron MLP to predict 3 observables from 5 input parameters (~5000 in training set)
- MLP outperforms other ML choices
- Deployed on nanoHUB for **education** (an attractive use of surrogates so students get answers fast)
- GE uses similar approach to give interactive Engine design options (200 in training set)

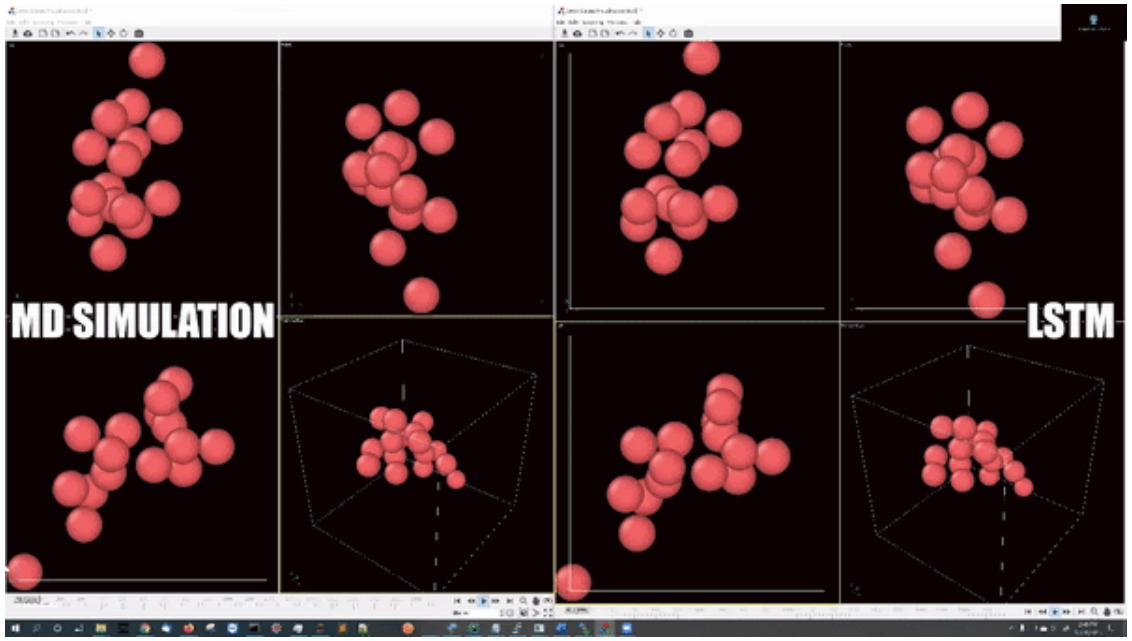
Direct simulation compared to Surrogates

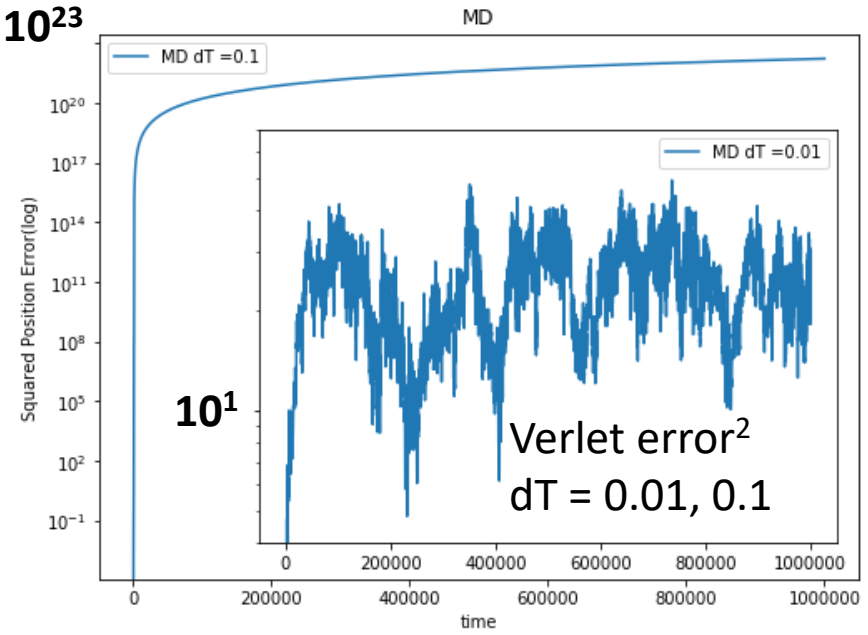


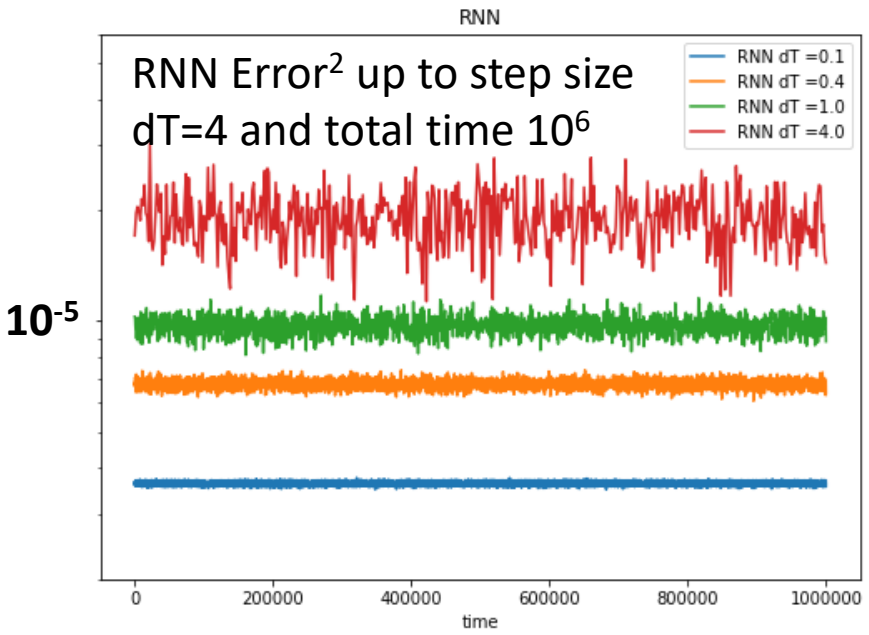
$$\text{Effective Speedup } S = \frac{T_{seq}(N_{lookup} + N_{train})}{T_{lookup}N_{lookup} + (T_{train} + T_{learn})N_{train}} \rightarrow 10^5 \text{ as } N_{lookup} \rightarrow \infty$$

Learn Newton's laws with Recurrent Neural Networks

- **Deep Learning is revolutionizing (spatial) Time series Analysis**
- Good example is integrating sets of differential equations
- Train the network on traditional 5 time step series from (Verlet) difference equations
- Verlet needs time step .001 for reliable integration but
- Learnt LSTM network is reliable for time steps which are **4000 times longer** and also learn potential.
- **Speedup is 30000** on 16 particles interacting with Lennard-Jones potentials
- 2 layer-64 units per layer LSTM network: **65,072 trainable parameters**
- 5000 training simulations

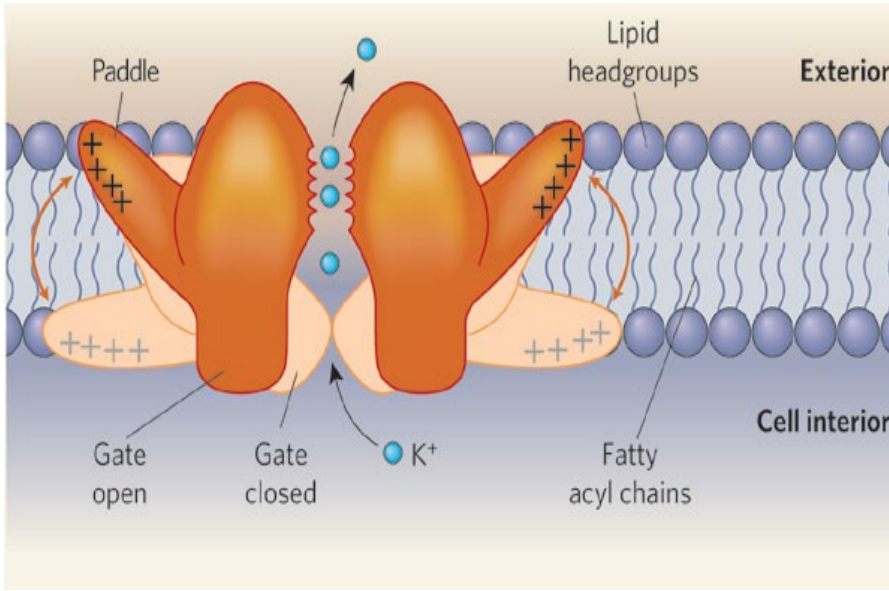


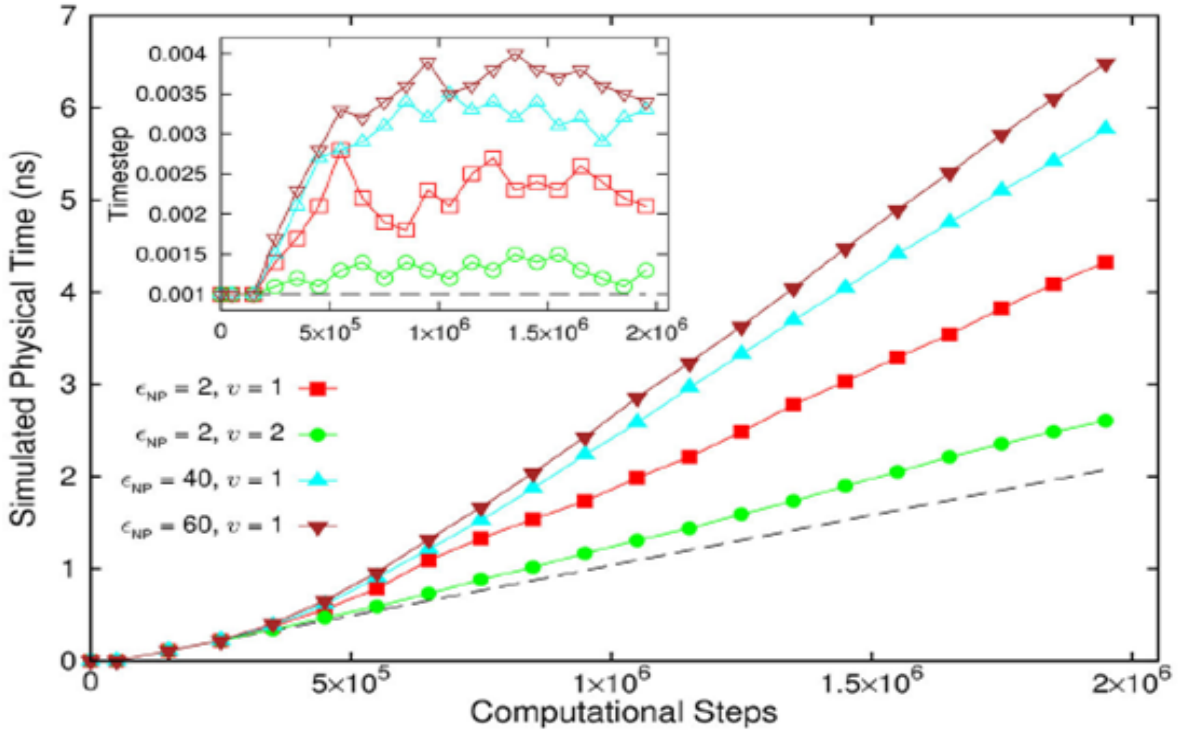




Dynamics of Ions near Polarizable Nanoparticles

- Choose time step dynamically using ML
- Choose consistency parameter (94.3% success) with ML
- Speedup of 3 from **MLAutoTuning** and a maximum speedup of 600 from the combination of ML and parallel computing.





Quality of simulation measured by time simulated per step with increasing use of ML enhancements. (Larger is better).

Inset is timestep used

Futures of MLaroundHPC (ML for CI) I

- **MLaroundHPC** broadly applicable but current **use is nonuniform** across domains
  - We need to improve Cyberinfrastructure support to make MLforHPC more effective for **more users**
- Use of modest DL network to **map material/potential drug structure to properties** (generalized QSAR) with simulation and observation: Advanced Progress
- Learn **surrogates** for **large scale simulations**: initial small scale results
- Use of MLforHPC in **agent-based systems** (learn agents replaing by surrogates): Very promising but few results
  - Use in Sociotechnical simulations and in virtual tissues (agents are people or cells)
- **Macroscopic** Structure as in learn complex multi-particle **potentials** scaling to N^7: many great successes
- **Learn Collective coordinates and guide ensemble** computations: dramatic progress with speedups up to 10^8
- **Microscale**; learn dynamics of small scale such as clouds, turbulence: Interesting results but much more to do
- Use of **Recurrent NN**'s to represent dynamics (**learn numerical differential operators**): Promising but only studied in small problems

Futures of MLaroundHPC (ML for CI) II

- Learn **errors** as well as values in differential equation solutions
- Look at **advanced solver** methods such as fast multipole or ML learnt potentials
- Quantify Deep Learning for (geospatial) Time Series for **different applications** and different methods: Looking at
  - Integration of ODE – apply surrogates to virtual cell-based simulations
  - Snow layers from polar science radar data
  - Earthquake Forecasting (are there actually hidden variables as phase transition?)
  - Industry is Ride Hailing and Transportation as well as audio
  - MLPerf has a Deep Learning for Time Series working group
- Investigate **different methods**: RNN (LSTM, GRU), RNN-T, non-recurrent with attention: Transformer
- Build **Software systems** to support AI dynamically mixed with simulation
- Investigate **optimized hardware** – CPU, Accelerator, Storage, Network for AI dynamically mixed with simulation
- How does this **hardware and software compare** to that for
  - Edge to Cloud data-center use case
  - Simulation supercomputer
  - Classic data-center big data problem

HPCforML: Similar Challenges in Parallelism for Big Data and Simulation

Complexity of Synchronization and Parallellization

Increasing Data

Loosely Coupled

Regular Coupling

Complex Coupling

Commodity Clouds

HPC Clouds: Accelerators
High Performance Interconnect

HPC Clouds/Supercomputers
Memory access also critical

MapReduce as in scalable databases

Global Machine Learning e.g. parallel clustering

Deep Learning

LDA

Graph Analytics e.g. subgraph mining

Pleasingly Parallel
Often independent events

Parameter sweep simulations

Regular simulations

Unstructured Adaptive Sparse

Structured Adaptive Sparse

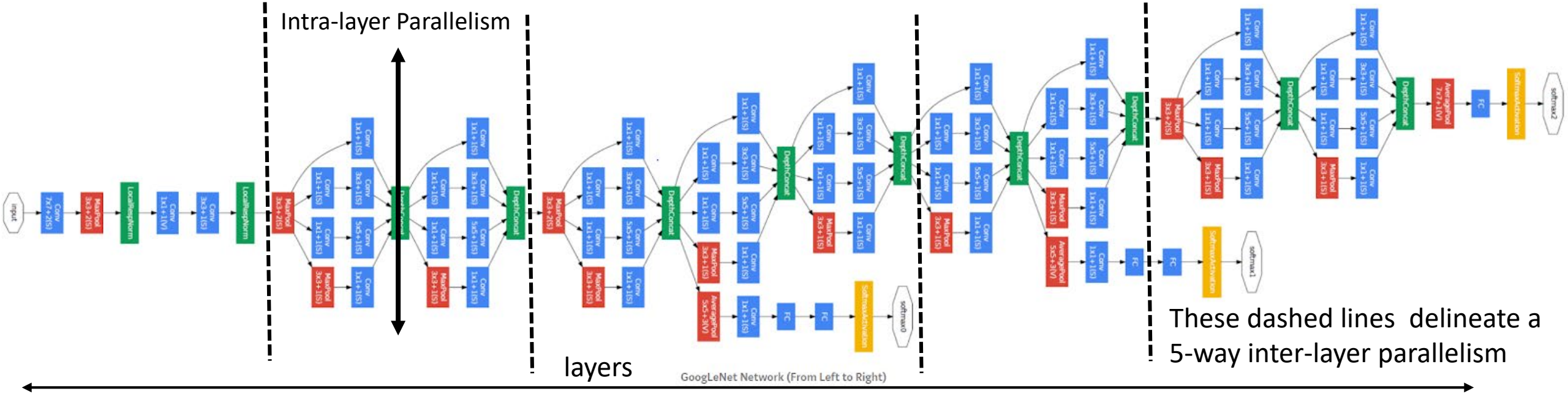
Current major Big Data category

Linear Algebra at core
(often not sparse)
Straightforward Parallelism

User Performed Parallelism

Simulations

HPCforML: The 4 Forms of Parallel Deep Learning are Fun!

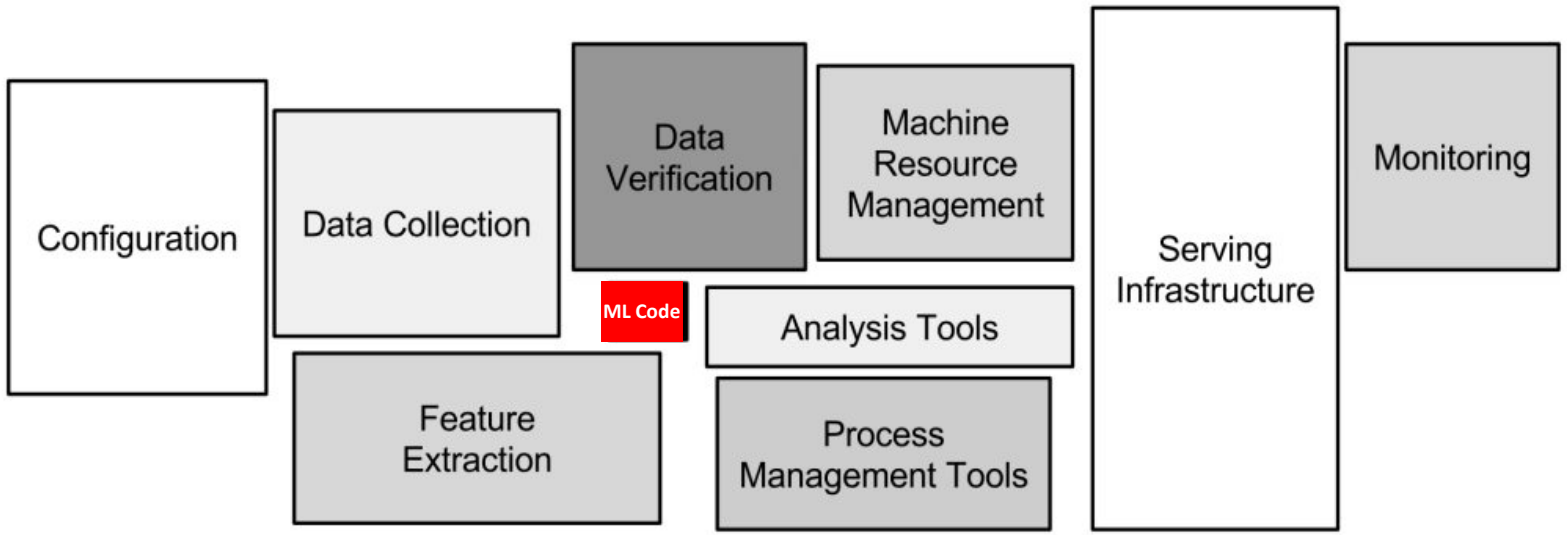


- **Data-parallelism:** Decompose tensor index corresponding to data into blocks i.e. divide the mini-batch into micro-batches on parallel nodes
- **Intra-layer model parallelism:** Decompose one or more other tensor indices
- **Inter-layer model (pipeline) parallelism:** Decompose model by groups of layers in the model
- **Hyper-parameter search parallelism** often with genetic algorithm
- All need to be efficiently integrated into system

HPCforML: Integration Challenges

Hidden Technical Debt in Machine Learning Systems

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips
{dsculley,gholt,dgg,edavydov,toddphillips}@google.com
Google, Inc.



"Only a fraction of real-world ML systems is composed of ML code"

This well-known paper points out that parallel high-performance machine learning is perhaps most fun but just a part of system. We need to integrate in the other data and orchestration components.

This integration is not very good or easy partly because data management systems like Spark are JVM-based which doesn't cleanly link to C++, Python world of high-performance ML

Twister2 at IU addresses this problem

NIPS 2015 http://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf