Accelerated Deep Learning Advances in HPC

William M. Tang

Princeton University/Princeton Plasma Physics Laboratory (PPPL)

GTC-DC-2017
Invited Talk #DC-7243
Location: Hemisphere A

Washington, DC

November 2, 2017

Co-authors: Julian Kates-Harbeck (Harvard U/PPPL), Alexey Svyatkovskiy (Princeton U), Kyle Felker (Princeton U/PPPL), and Eliot Feibush (PPPL/Princeton U)

Background/Approach

- Deep Learning Method: distributed data-parallel approach to train deep neural networks → *Python framework using high-level Keras library with Google Tensorflow backend*

-- major contrast with "Shallow Learning" approaches including SVM's, Random Forests, Single Layer Neural Nets, etc. – including (i) deployment of accelerators (e.g., modern GPU's); and (ii) move from DL software deployment on clusters to supercomputers:

→ *Titan (ORNL), Summit-Dev (ORNL); Piz Daint (CSCS);*
Tsubame-3 (TiTech) + Intel Systems – Cori (LBNL), Theta (ANL)

-- stochastic gradient descent (SGD) used for large-scale (i.e., optimization on supercomputers) with parallelization via mini-batch training to reduce communication costs

- DL Supercomputer Challenge: scaling studies to examine if convergence rate saturates with increasing mini-batch size (to thousands of GPU's)

APPLICATION TOPIC: FUSION ENERGY SCIENCE

SITUATION ANALYSIS

Most critical problem for Fusion Energy: *avoid/mitigate large-scale major disruptions*

•Approach: Use of big-data-driven statistical/machine-learning (ML) predictions for the occurrence of disruptions in EUROFUSION facility “Joint European Torus (JET)”

•Current Status: ~ 8 years of R&D results (led by JET) using Support Vector Machine (SVM) ML on *zero-D* time trace data executed on CPU clusters yielding ~ *reported success rates in mid-80% range for JET 30 ms before disruptions , BUT > 95% with false alarm rate < 3% actually needed for ITER* (Reference – P. DeVries, et al. (2015))

•**Princeton Team Goals include:**

(i)improve physics fidelity via development of new *ML **multi-D**, time-dependent software including better classifiers;*

(ii)develop ***“portable”*** (cross-machine) predictive software beyond JET to other devices and eventually ITER; and

(iii)enhance execution speed of disruption analysis for very large datasets

→ *development & deployment of advanced ML software via **Deep Learning Recurrent Neural Networks***

CLASSIFICATION

- Binary Classification Problem:
  - *Shots are Disruptive or Non-Disruptive*
- ***Supervised*** ML techniques:
  - *Physics domain scientists combine knowledge base of observationally validated information with advanced statistical/ML predictive methods.*
- Machine Learning (ML) Methods Engaged:

  Basic SVM approach initiated by JET team producing “APODIS” software *leading* **now to** *Princeton’s New Deep Learning Fusion Recurrent Neural Net (FRNN) code*

- Approach: (i) *examine appropriately normalized data;* (ii) *use training set to generate model;* (iii) *use trained model to classify new samples*

  → Multi-D data analysis requires new signal representations;

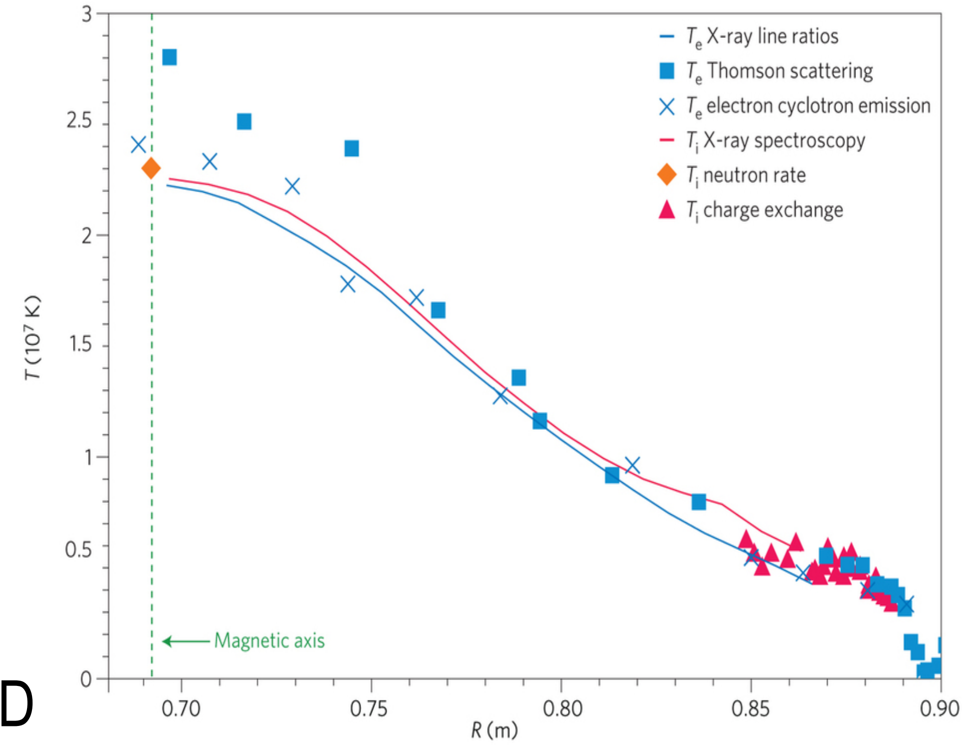
  → FRNN software includes *Deep Learning Convolutional and Recurrent Neural Net features*.

Challenges & Opportunities

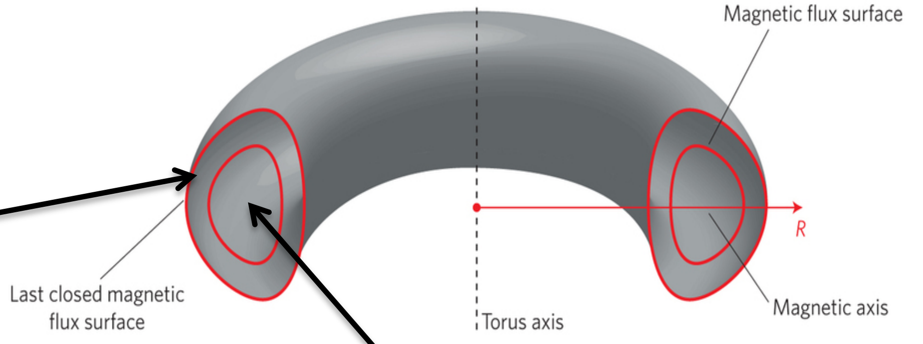
Higher Dimensional Signals

- At each timestep: **arrays** instead of **scalars**
- All as a function of ***ρ*** (normalized flux surface)
- Raw 1D profile → convolution, optimize pooling for most salient features
- Full feature vectors/arrays include zero-D plus 1D

- Examples:
  - –1D Current profiles
  - –1D Electron temperature profiles
  - –1D Radiation profiles



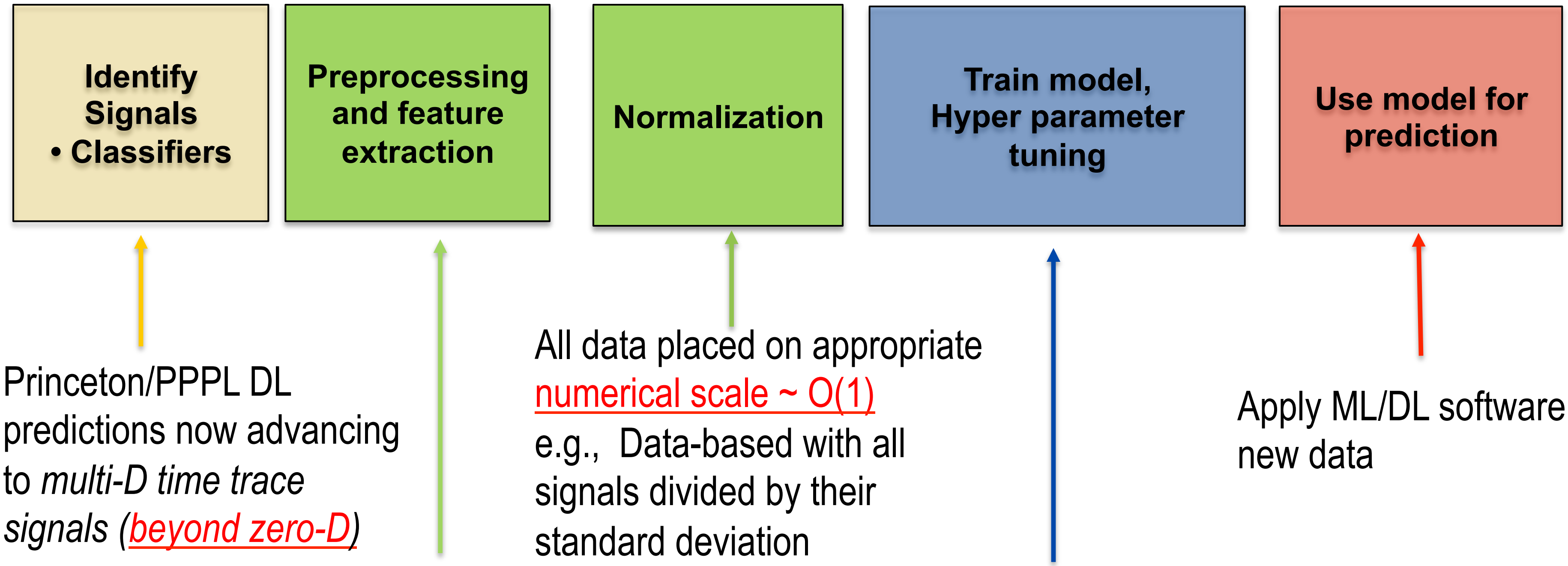
ρ = 1



ρ = 0

Mazon, Didier, Christel Fenzi, and Roland Sabot. "As hot as it gets." Nature Physics 12.1 (2016): 14-17.

Machine Learning Workflow



Identify Signals
• Classifiers

Preprocessing and feature extraction

Normalization

Train model, Hyper parameter tuning

Use model for prediction

Princeton/PPPL DL predictions now advancing to *multi-D time trace signals* (*beyond zero-D*)

Measured sequential data arranged in patches of equal length for training

All data placed on appropriate numerical scale ~ O(1)
e.g., Data-based with all signals divided by their standard deviation

- All available data analyzed;
- Train LSTM (Long Short Term Memory Network) iteratively;
- Evaluate using ROC (Receiver Operating Characteristics) and cross-validation loss for every epoch (equivalent of entire data set for each iteration)

Apply ML/DL software on new data

JET Disruption Data

# Shots	Disruptive	Nondisruptive	Totals
Carbon Wall	324	4029	4353
Beryllium Wall (ILW)	185	1036	1221
Totals	509	5065	5574

JET produces ~ Terabyte (TB) of data per day

JET studies → 9 Signals of zero-D (scalar) time traces, including	Data Size (GB)
Plasma Current	1.8
Mode Lock Amplitude	1.8
Plasma Density	7.8
Radiated Power	30.0
Total Input Power	3.0
d/dt Stored Diamagnetic Energy	2.9
Plasma Internal Inductance	3.0

~55 GB data collected from each JET shot

→***Well over 350 TB total amount with multi-dimensional data yet to be analyzed***

Deep Recurrent Neural Networks (RNNs): Basic Description

- “Deep”
  - **Learn salient representation** of complex, *higher dimensional* data

- “Recurrent”
  - Output **h**(t) depends on input **x**(t) & internal state **s**(t-1)

Internal State (“memory/context”)

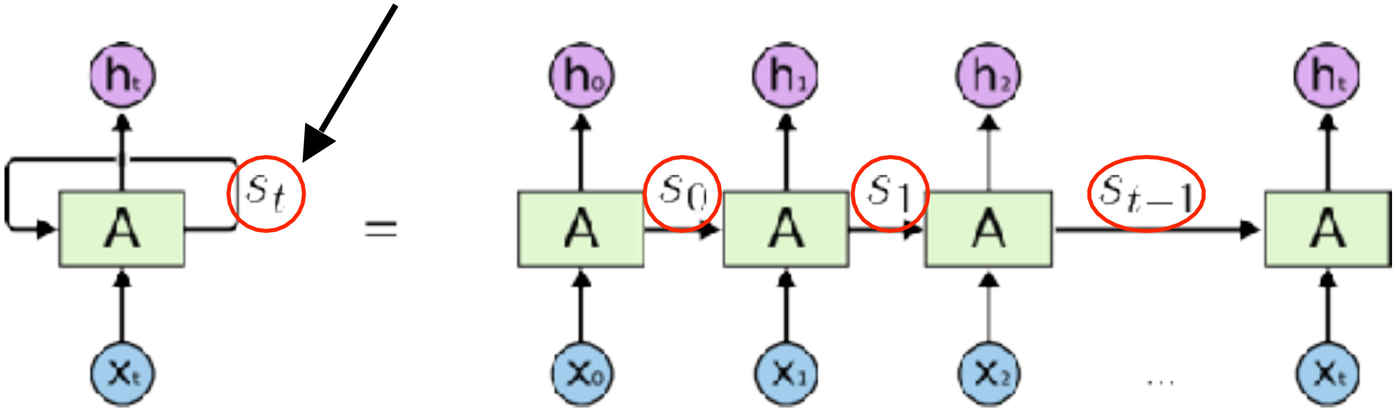


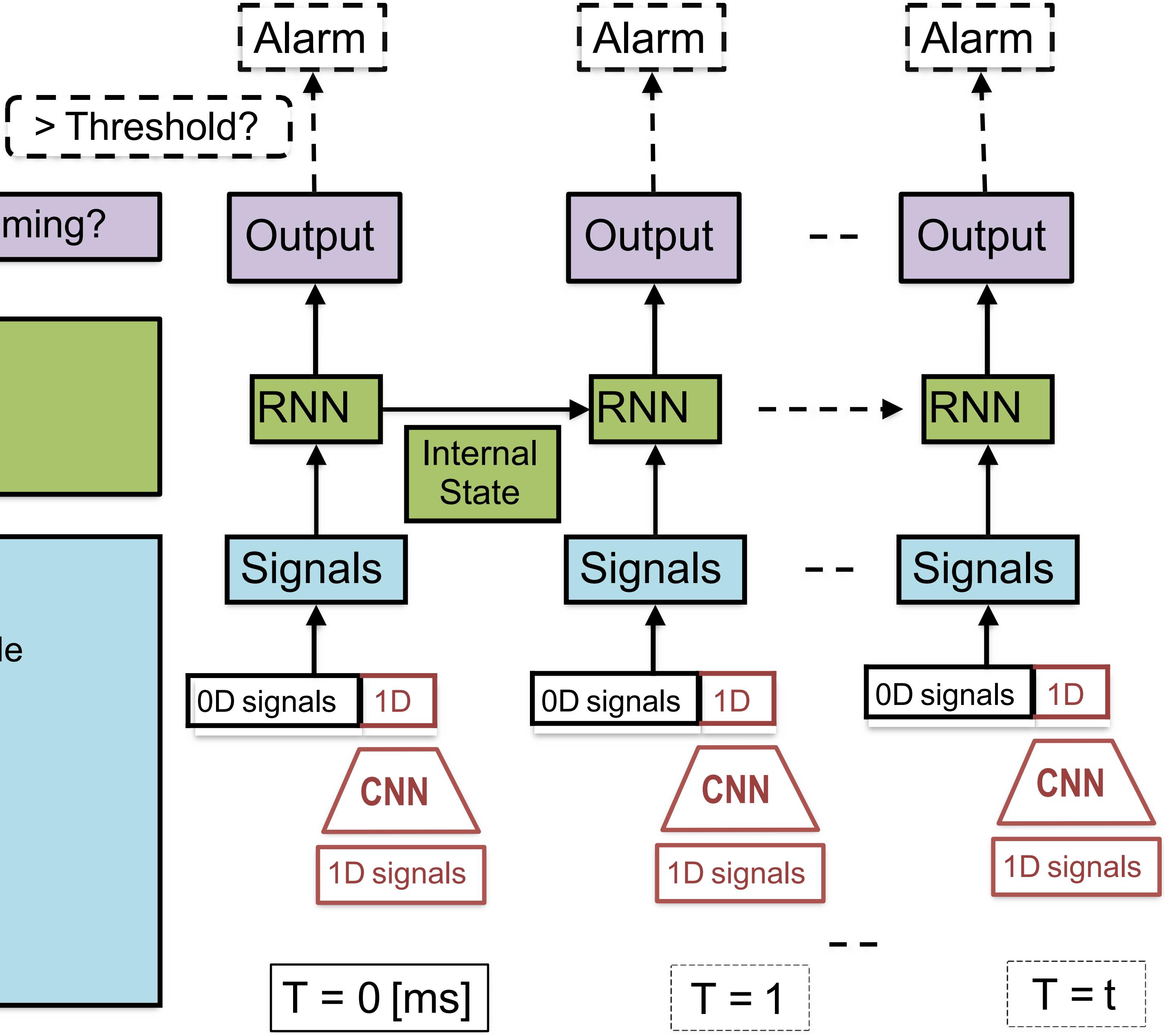
Image adapted from: colah.github.io

Deep Recurrent Neural Nets: Schematic

Output: Disruption coming?

RNN Architecture:
- LSTM
- 3 layers
- 300 cells per layer

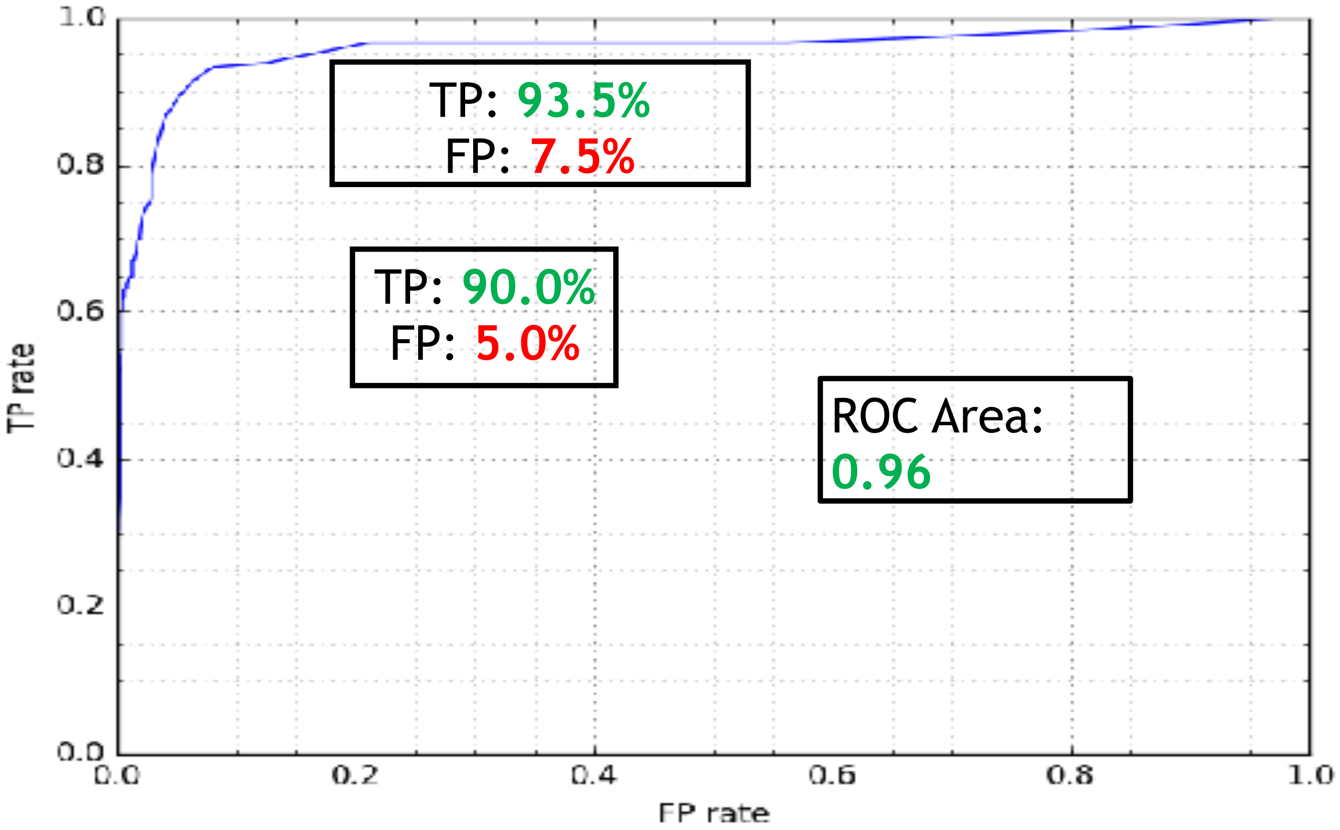
Signals:
- Plasma Current
- Locked Mode Amplitude
- Plasma Density
- Internal Inductance
- Input Power
- Radiated Power
- Internal Energy
- 1D profiles (electron temperature, density)
- …



FRNN Code PERFORMANCE: ROC CURVES

JET ITER-like Wall Cases @30ms before Disruption

Performance Tradeoff: Tune True Positives (good: correctly caught disruption) **vs. False Positives** (bad: safe shot incorrectly labeled disruptive).



TP: 93.5%
FP: 7.5%

TP: 90.0%
FP: 5.0%

ROC Area:
0.96

Data (~50 GB), 0D signals:

- Training: on **4100 shots** from **JET C-Wall** campaigns
- Testing **1200 shots** from **Jet ILW** campaigns
- **All shots used**, no signal filtering or removal of shots

RNNs: HPC Innovations Engaged

GPU training

●Neural networks use dense tensor manipulations, efficient use of GPU FLOPS

●Over 10x speedup better than multicore node training (CPU's)

Distributed Training via MPI

Linear scaling:

●Key benchmark of "time to accuracy": we can train a **model that achieves the same results nearly N times faster with N GPUs**

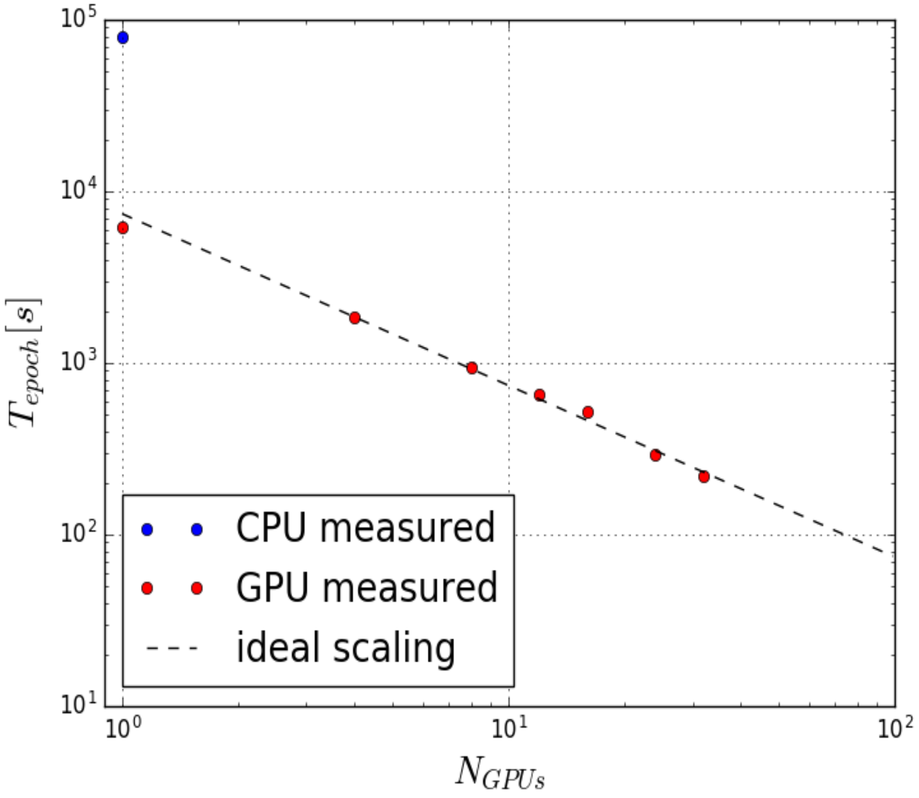
Scalable

●to 100s or >1000's of GPU's on Leadership Class Facilities

●TB's of data and more

●Example: Best model training time on full dataset (~40GB, 4500 shots) of 0D signals training

- ○ **SVM** (JET) **: > 24hrs**
- ○ **RNN** (20 GPU's) : **~40min**



Fusion Recurrent Neural Net (FRNN) Description

- **Python deep learning code** for disruption prediction in fusion (tokamak) experiments
  - Reference: ***https://github.com/PPPLDeepLearning/plasma-python***
- Implements distributed data parallel synchronous RNN training
  - *Tensorflow & Theano backends with MPI for communication*
  - FRNN code workflow is characteristic of typical distributed deep learning software
  - Core modules:
    - **Models:** Python classes necessary to construct, train, and optimize deep RNN models.
    - **Pre-process:** arrange data into patches for stateful training; normalize
    - **Primitives:** Python objects for key plasma physics abstractions
    - **Utils:** a set of auxiliary functions for pre-processing, performance evaluation, and learning curves analysis

Scaling Summary

Communication: each batch of data requires time for synchronization

$$T_{sync} \sim log\left(N_{workers}\right)$$

Runtime: computation time

$$T \sim \frac{1}{N}\left(A + B\ log(N)\right) = O\left(\frac{log(N)}{N}\right)$$

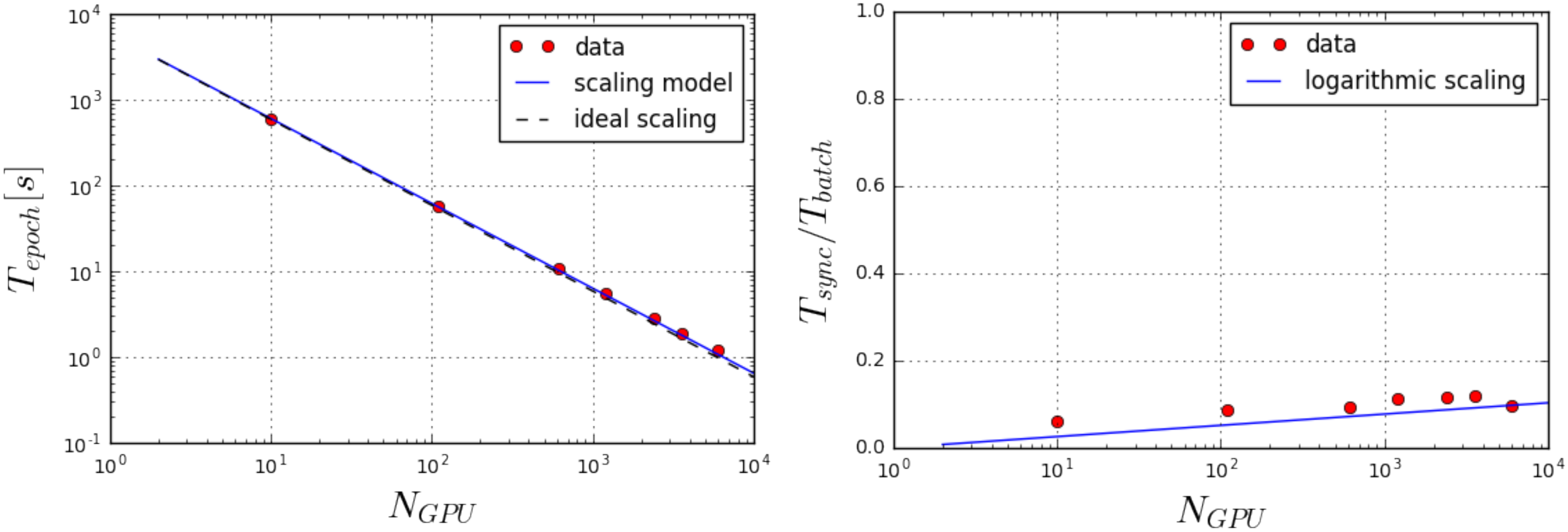
Parallel Efficiency

$$\text{Parallel Efficiency} \sim \frac{A + B}{A + B\ log(N)} = o\left(\frac{1}{log(N)}\right)$$

FRNN Scaling Results on GPU's

- Tests on OLCF Titan CRAY supercomputer
  - ***OLCF DD AWARD**: Enabled Scaling Studies on Titan currently up to 6000 GPU's*
  - Total ~ 18.7K Tesla K20X Kepler GPUs

Tensorflow+MPI



CURRENT PERSPECTIVE

Forecasting disruptions using machine learning is an important application of a **general idea:**

→ Use multi outcome prediction to **distinguish disruption types/scenarios**

→ Beginning now to move from ***prediction to active control*** (including new experimental proposals on the U.S. DIII-D tokamak in San Diego, CA)

→ Increasingly large and diverse data sets require building *scalable systems to take advantage of leadership class computing facilities*

Fusion Deep Learning (FRNN) Technical Summary

- FRNN → a distributed data-parallel approach to train deep neural networks (stacked LSTM's);
- Replica of the model is kept on each "worker" → *processing different mini-batches of the training dataset in parallel;*
- Results on each worker are combined after each *epoch* using MPI;
- Model parameters are synchronized via parameter averaging → *with learning rate adjusted after each epoch to improve convergence*
- **Stochastic gradient descent (SGD)** used for large-scale optimization with parallelization via mini-batch training to reduce communication cost.

→ **Challenge**: scaling studies to examine if *convergence rate saturates/ decreases with increasing mini-batch size (to thousands of GPU's).*

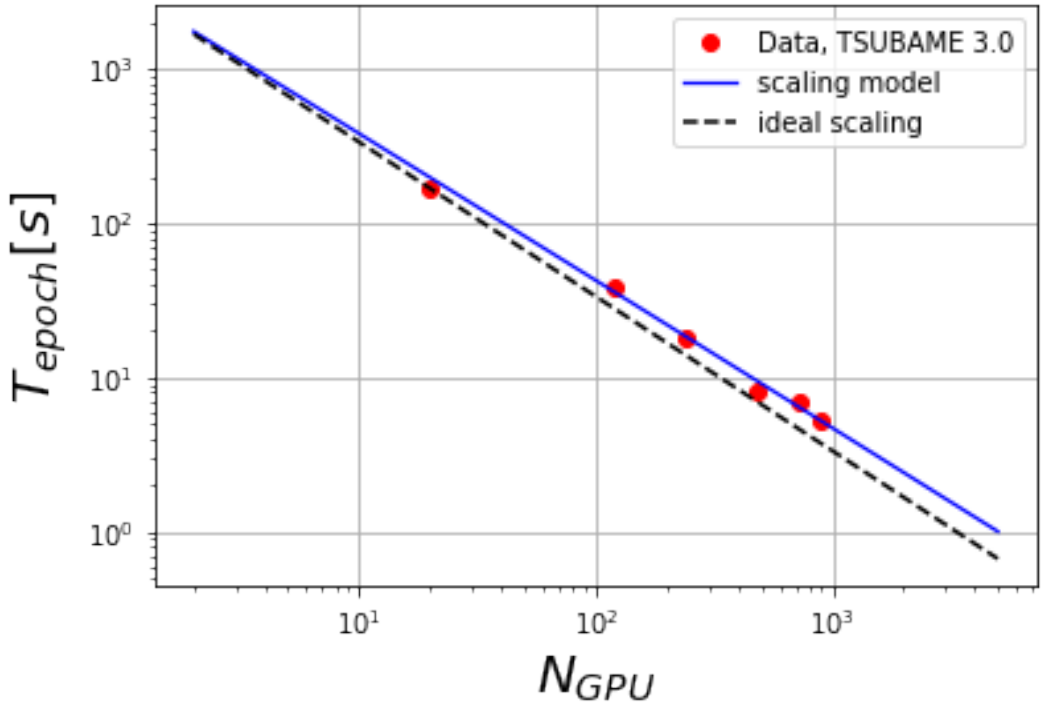
→ **Targeted Large HPC Systems with P-100's for Performance Scaling Studies:** (1) "TSUBAME 3" @ TITECH with ~ 3K GPU's – "Grand Challenge Runs"; (2) "PIZ-DAINT" Cray XC50 @ CSCS (Switzerland) with > 4K GPU'S; (3) "SUMMIT-DEV" @ OLCF leading to SUMMIT with VOLTA GPU's

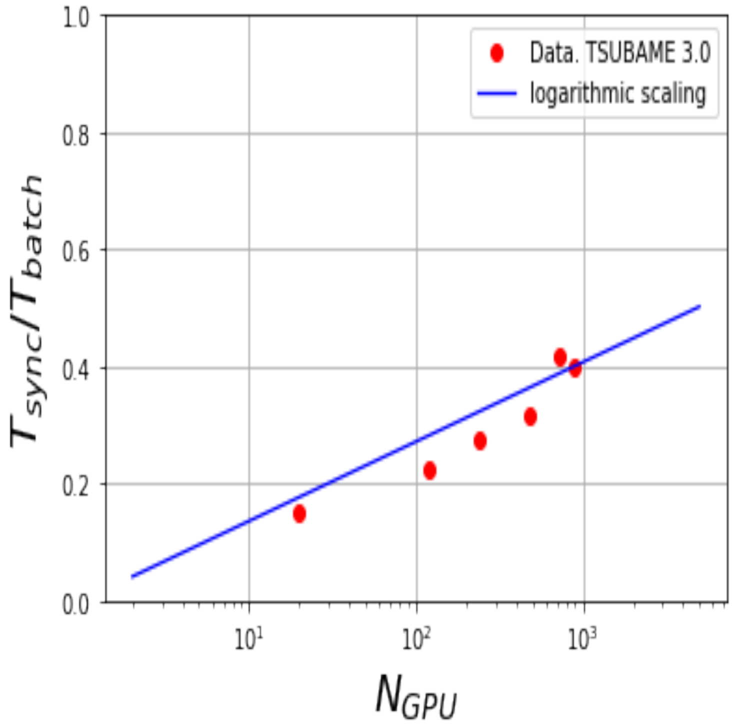
New FRNN scaling tests: TSUBAME 3.0

Very recent results: TSUBAME 3.0 supercomputer (TiTech, Tokyo, Japan)

Tsubame 3.0 initial “Grand Challenge Runs”

- *Order of thousand Tesla P100 SXM2 GPUs, 4 GPUs per node, NVlink*
- *Tensorflow+MPI, CUDA8, CuDNN 6, OpenMPI 2.1.1, GPU Direct*





Fusion Deep Learning (FRNN) Technical Summary (continued)

NVIDIA Volta GPU's ➔ to be key element of 200 PF SUMMIT @ OLCF

Associated Challenge: requires training neural networks with "half-precision floats"
- Single-Precision → 32 bits (8 bits for exponent, 23 for fraction and 1 for sign)
- Double-Precision → 64 bits

NOTE: FRNN code has produced many results with single precision - float32 and has now developed new half-precision float - 5 bits exponent, 10 bit fraction and 1 bit sign

REFERENCE: *half-precision float deployment of FRNN with cross-benchmarking of new results vs. earlier single precision results → paper to be presented at SC'17 (Denver, CO)– includes description of changes in the weight update during SGD (Stochastic Gradient Descent) method to prevent vanishing gradients due to lower precision*.

- *Looking forward to testing new half-precision FRNN software capability on NVIDIA Volta GPU's at OLCF*

Fusion Deep Learning (FRNN) Technical Summary (continued)

→ GOOGLE Article on Tensor Processing Units in Cloud:

"Build and Train Machine Learning Models on our new Google Cloud TPU's"
(Tensor Processing Units)
https://blog.google/topics/google-cloud/google-cloud-offer-tpus-machine-learning/

The highlighted description highlights ***potential delivery of 11.5 PF of compute power to expedite training!***

Possible FRNN Software Relevance:
Since FRNN software already uses the TensorFlow backend, our current plan is to try the *Google Cloud TPU's* -- beginning with their offer of free access to 1000 Cloud TPU's via the TensorFlow Research Cloud – for which we have applied.

→ APPROACH: *Comparison of time to prediction and associated deep learning neural nets training rates on supercomputers vs. that on the new Google Cloud TPU's promises to be quite informative.*

Fusion Big Data ML/DL Application Summary

- **Fusion Energy Mission**:

  -- *Accelerate demonstration of the scientific & technical feasiblity of delivering Fusion Power*

  -- *Most critical associated problem is to avoid/mitigate large-scale major disruptions.*

- **ML Relevance to HPC:**

  -- ***Rapid Advances*** *on development of predictive methods via large-data-driven "machine-learning" statistical methods*

  -- ***Approach Focus:*** *Deep Learning/Recurrent Neural Nets (RNNs)*

  -- ***Significance:*** *Exciting alternative predictive approach to "hypothesis-driven/first principles" exascale predictive methods*

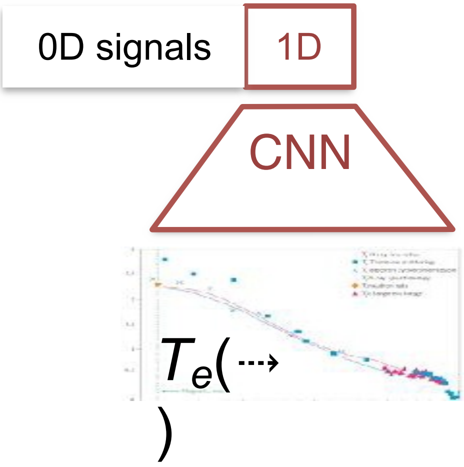
  *** ***Convergence/Complementarity:*** *Physics-centric path-to-exascale* ***HPC needed to introduce/establish improved Supervised ML Classifiers with associated features***

- **Associated Challenge:**

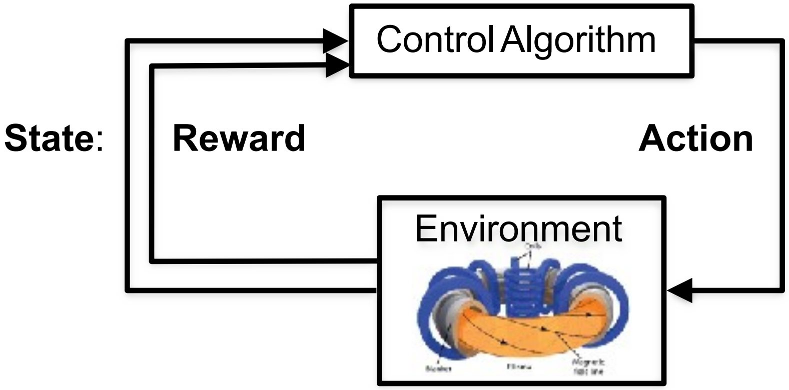
  → *Improvements over zero-D SVM-based machine-learning needed to achieve > 95% success rate, <5% false positives at least 30 ms before disruptions -- with portability of software to ITER via enhanced physics fidelity (capturing multi-D) with improvement in execution time enabled by access to advanced HPC hardware (e.g., large GPU and possibly other supercomputing systems).*

Takeaways: Deep Learning Analysis

Use **Higher-dimensional signals**

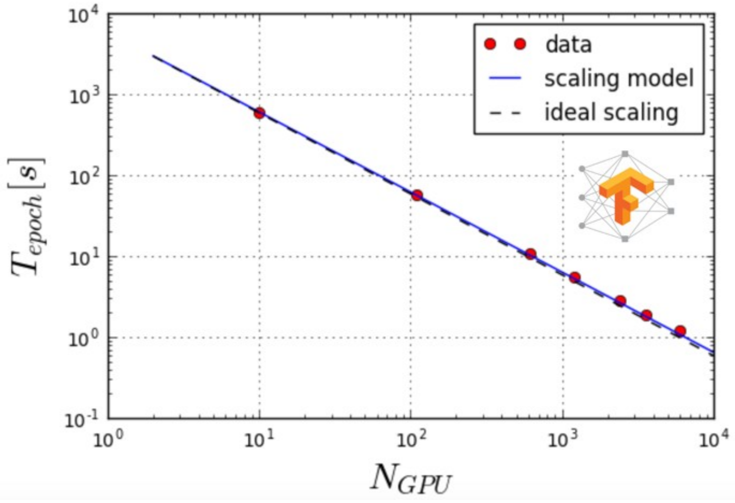


Automatically learn **cross-machine, generalizable features**



Go from **prediction to control (deep reinforcement learning)**

Take advantage of **world class HPC**


10^4
10^3
10^2
10^1
10^0
10^{-1}
$T_{epoch}\,[s]$
10^0
10^1
10^2
10^3
10^4
N_{GPU}
● ● data
— scaling model
- - ideal scaling