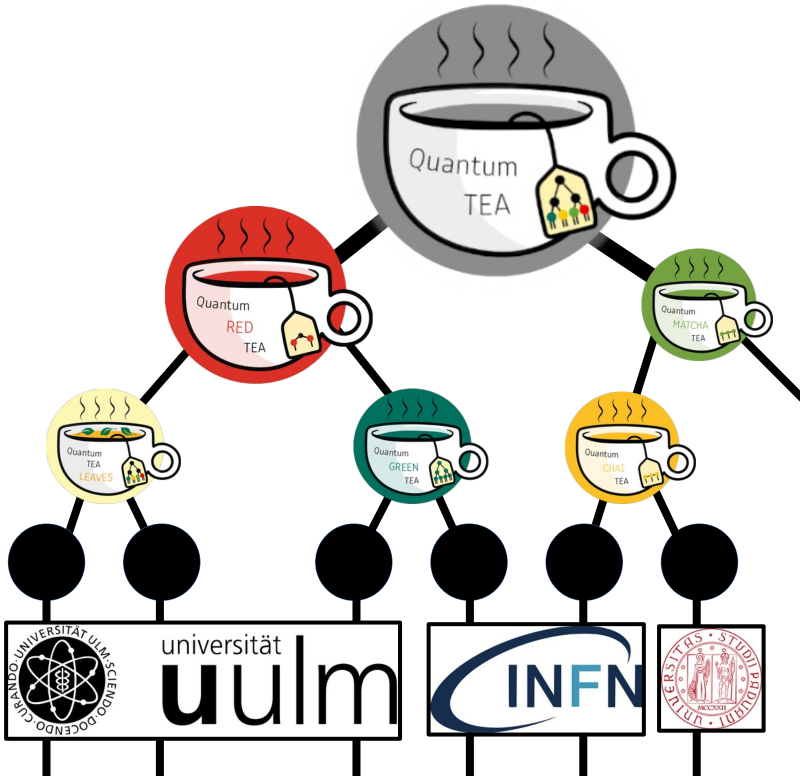
Quantum-inspired tensor-network machine learning: finding optimal hyperparameters, libraries, and hardware

Daniel Jaschke, Alberto Coppi, Marco Ballarin, and Simone Montangero

AI @ INFN

Logo by N. Reinic



Outline today's "Tensor network machine learning"

Supervised machine learning

Unsupervised / Generative

Neural Networks & Deep learning

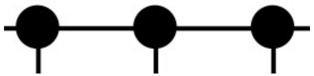
Quantum-inspired methods

Quantum Machine Learning (QML)

What are “quantum-inspired” methods?

Method or idea for quantum technology used on a classical problem without a QPU

Method: tensor network (TN) algorithms



- Quantum wave function represented by TN
- Compress entanglement (quantum correlations)

Problem: supervised learning

- Model represented by TN
- Compress information

What are “quantum-inspired” methods?

Method or idea for quantum technology used on a classical problem without a QPU

Method: tensor network (TN) algorithms

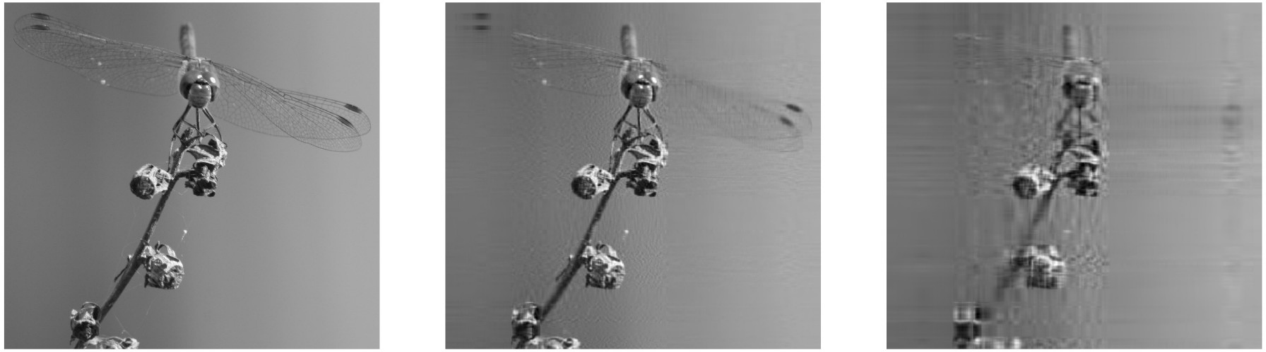
- Quantum wave function represented by TN
- Compress entanglement (quantum correlations)

Problem: supervised learning

- Model represented by TN
- Compress information

Compression example
Singular Value Decomposition

Singular values kept:
1549 versus 50 versus 15
(1549: no compression)



Quantum TEA: where do we come from?

Quantum Tensor network **E**mulator **A**pplications

Can I just have tensor networks for quantum information?

How to emulate a digital quantum circuits?

+

Quantum TEA: where do we come from?

Quantum Tensor network **E**mulator **A**pplications

Can I just have tensor networks for quantum information?

How to emulate a digital quantum circuits?

+

How to solve the Schrödinger equation?

+

Can we do tensor network machine learning? … soon as well public.

+

Quantum TEA: where do we come from?

Quantum **T**ensor network **E**mulator **A**pplications

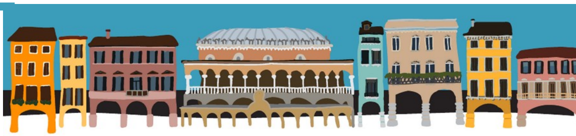
Can I just have tensor networks for quantum information?

How to emulate a digital quantum circuits? +

How to solve the Schrödinger equation? +

And another for performance

Can we do tensor network machine learning? … soon as well public. +



The one-pixel example

Data set: Training & **test**	Feature map	Guess TN	Optimization	Inference
Value: 220 Label: 0	$\begin{pmatrix}\cos(220/255)\\ \sin(220/255)\end{pmatrix}$			
Value: 18 Label: 1	$\begin{pmatrix}0.9975\\ 0.0705\end{pmatrix}$			
Value: 34 Label: 1	$\begin{pmatrix}0.9911\\ 0.1329\end{pmatrix}$			

The one-pixel example

Data set: Training & test

Value: 220
Label: 0

Value: 18
Label: 1

Value: 34
Label: 1

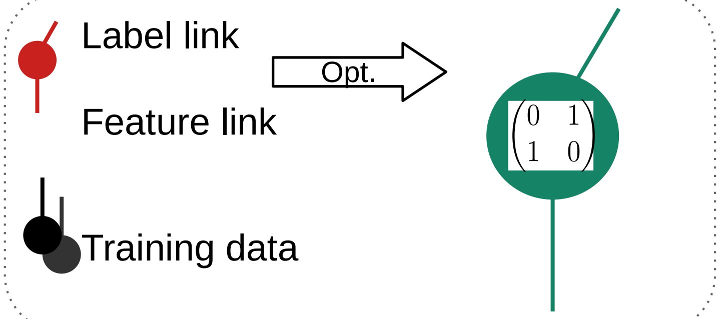
Feature map

$\begin{pmatrix} \cos(220/255) \\ \sin(220/255) \end{pmatrix}$

$\begin{pmatrix} 0.9975 \\ 0.0705 \end{pmatrix}$

$\begin{pmatrix} 0.9911 \\ 0.1329 \end{pmatrix}$

Guess TN / **Optimization**



$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

Inference

$\begin{pmatrix} 0.99 \\ 0.07 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

$\text{argmax} \begin{pmatrix} 0.07 \\ 0.99 \end{pmatrix} = 1$

The one-pixel example

Data set: Training & **test**	Feature map	Guess TN	Optimization	Inference

Value: 220
Label: 0

$\begin{pmatrix} \cos(220/255) \\ \sin(220/255) \end{pmatrix}$

Value: 18
Label: 1

$\begin{pmatrix} 0.9975 \\ 0.0705 \end{pmatrix}$

Value: 34
Label: 1

$\begin{pmatrix} 0.9911 \\ 0.1329 \end{pmatrix}$

Label link

Feature link

Training data

Opt.

$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

$\begin{pmatrix} 0.99 \\ 0.07 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

$\operatorname{argmax} \begin{pmatrix} 0.07 \\ 0.99 \end{pmatrix} = 1$

$= \begin{pmatrix} 0.13 \\ 0.99 \end{pmatrix}$

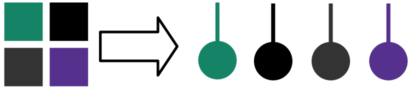
argmax 1

Full-scale algorithm

Based on / original work: E. Miles Stoudenmire and David J. Schwab, Supervised Learning With Quantum-Inspired Tensor Networks, arXiv:1605.05775v2 (2017)

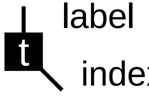
Feature map: list of vectors

One sample: (2x2 pixels)

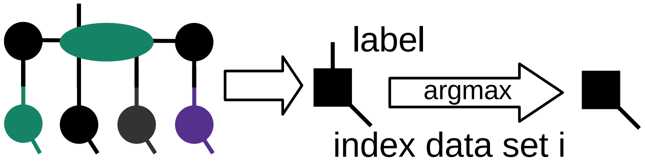


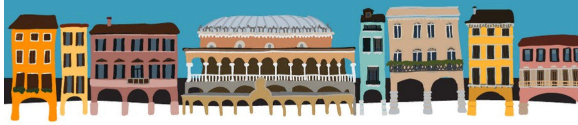
Outer product of vectors (many-body state without quantum correlations)

True labels for a data set:



Inference



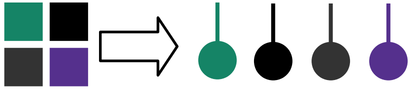


Full-scale algorithm

Based on / original work: E. Miles Stoudenmire and David J. Schwab, Supervised Learning With Quantum-Inspired Tensor Networks, arXiv:1605.05775v2 (2017)

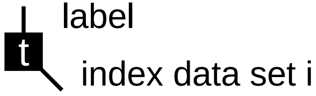
Feature map: list of vectors

One sample: (2x2 pixels)

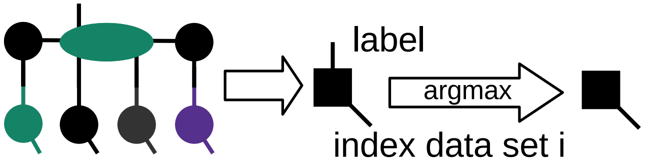


Outer product of vectors (many-body state without quantum correlations)

True labels for a data set:

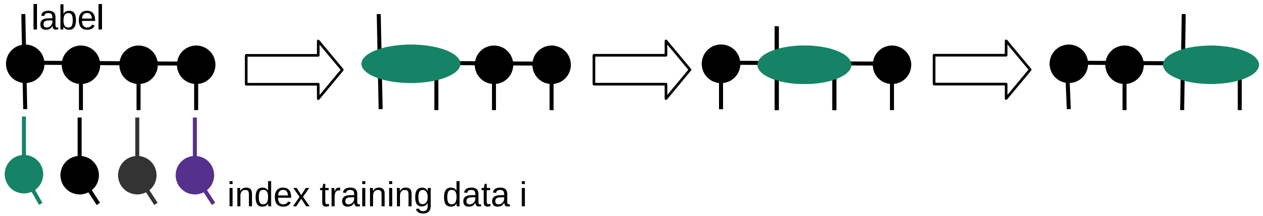


Inference



Optimization step: (no, no backpropagation & gradient descent)

→ One sweep optimizes each nearest-neighbor tensor-pair once

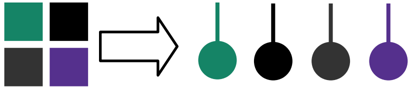


Full-scale algorithm

Based on / original work: E. Miles Stoudenmire and David J. Schwab, Supervised Learning With Quantum-Inspired Tensor Networks, arXiv:1605.05775v2 (2017)

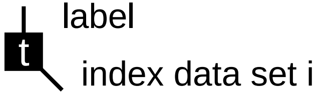
Feature map: list of vectors

One sample: (2x2 pixels)

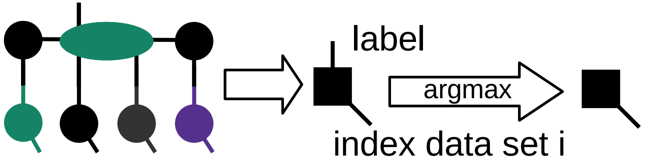


Outer product of vectors (many-body state without quantum correlations)

True labels for a data set:

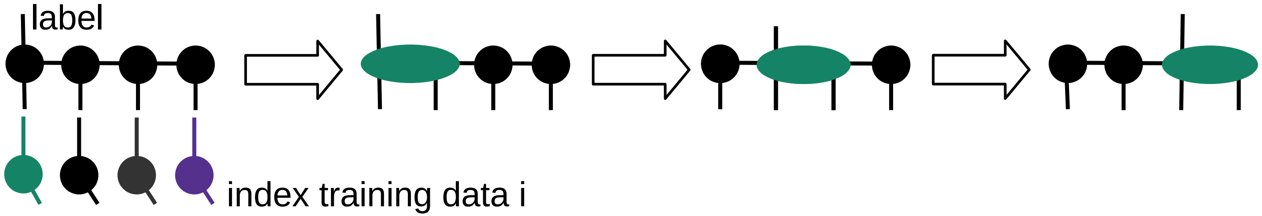


Inference

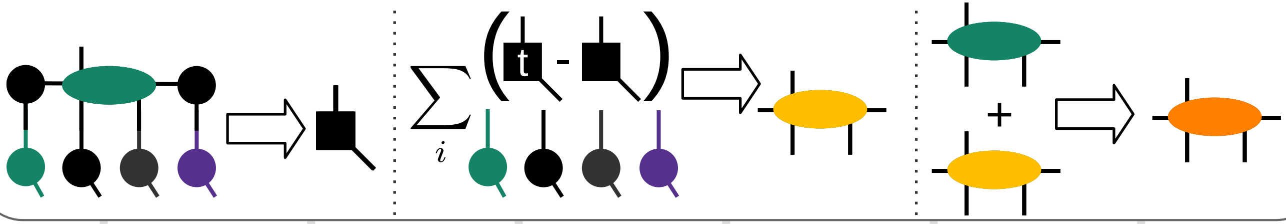


Optimization step: (no, no backpropagation & gradient descent)

→ One sweep optimizes each nearest-neighbor tensor-pair once



→ Calculate the local gradient for the two-site tensor



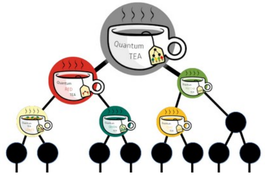
Hyper-parameters, libraries, and hardware

What do we want to bechmark / tune?

Ansätze

MPS

TTN



Parameters in ansatz:

- Bond dimension (compression)
- Precision: double, single, half
- Update 1- or 2-tensors together
- Mapping 2d image
- Label link or “pure overlap”

Libraries & hardware

- CPU versus GPU
- numpy/cupy versus torch

Machine:
leonardo booster partition with A100 GPU

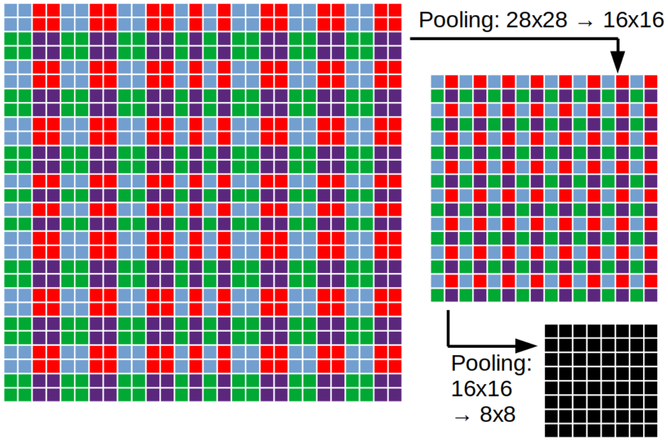
Reference point: MNIST

Training: 2,000
Training batch: 1000
Test: 2,000
Averaged: no (1 run)
MNIST: 70,000

Training precision: 96.2%
Test precision: 92.3% *
Walltime: 1372 seconds

$\chi = 32$

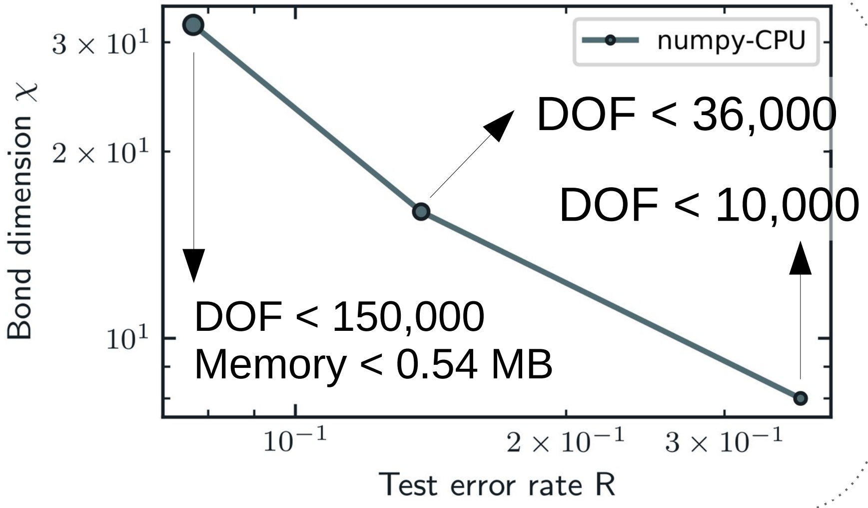
* and one can do better, see Stoudenmire & Schwab, arXiv:1605.05775v2 with > 99%

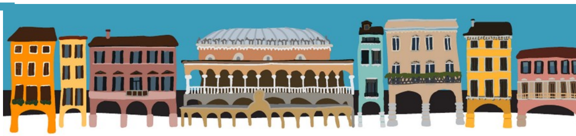


Bond dimension χ

- compression rate quantum correlations as quantum-inspired explanation

- Degrees of freedom (DOF) or memory-need or expressability

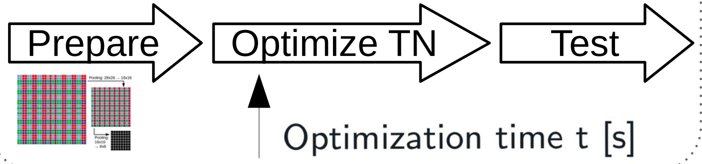


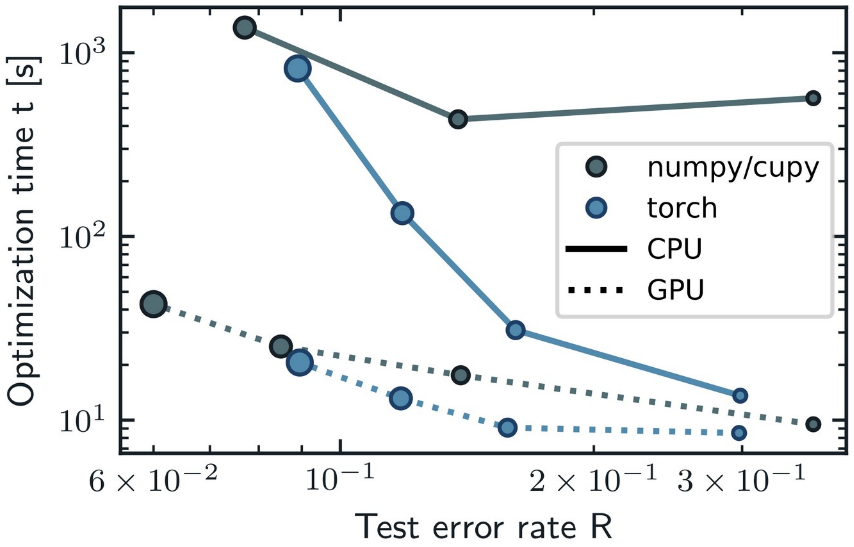


Numpy versus torch, CPU versus GPU

Torch: despite machine learning, no backpropagation used here

Time to run simulation

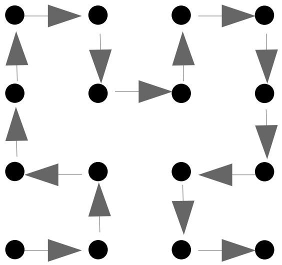




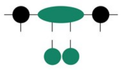
Continue with torch, GPU, and $\chi = 64$

2d image … how to get to a 1d system?

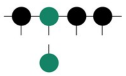
Hilbert curve



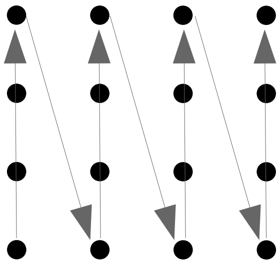
Test accuracy
91.05%



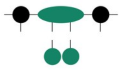
Test accuracy
91.05%



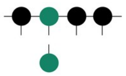
ZigZag map



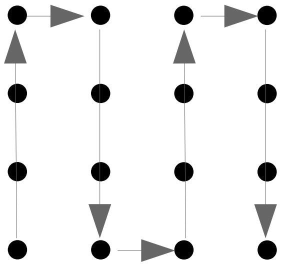
Test accuracy
91.35%



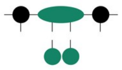
Test accuracy
91.45%



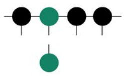
Snake map



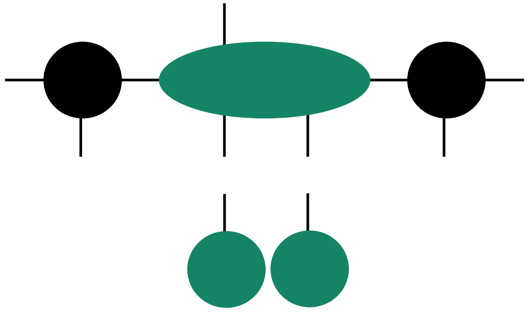
Test accuracy
90.9%

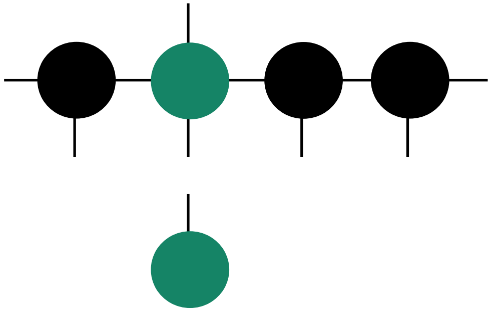


Test accuracy
91.15%



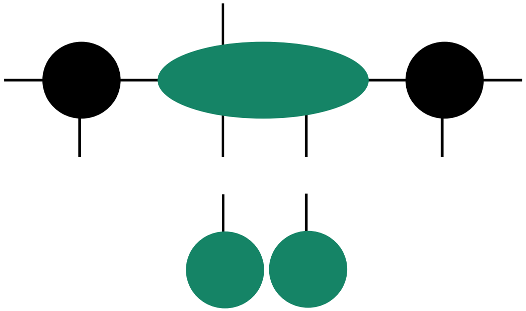
Single-tensor updates





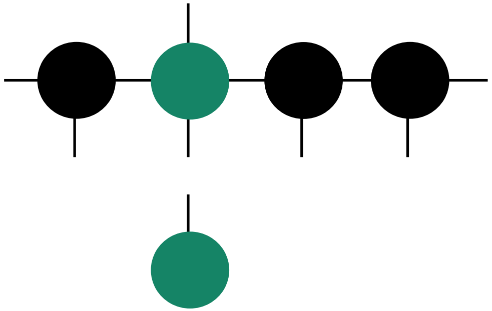
Single-tensor updates

Two-tensor update



$O(\chi^3 \cdot d^3 \cdot l), d = 2, l = 10$

Single-tensor update



$O(\chi^3 \cdot d^2 \cdot l), d = 2, l = 10$

Test accuracy: 91.05%
Time: 20.59s

Test accuracy: 91.05%
Time: 14.43s

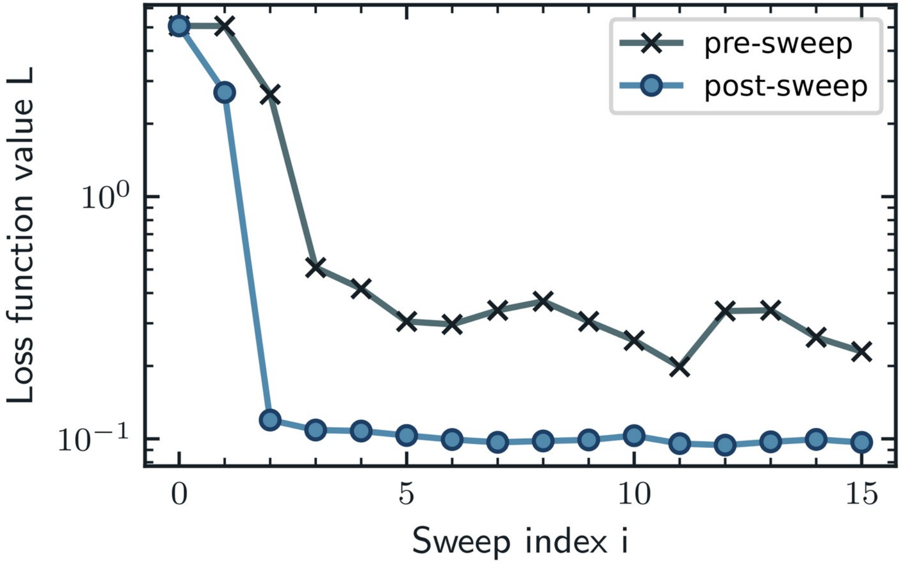
Apply lessons learned to the full data set

Single-tensor update

Training: 60,000
Training batch: 20,000
Test: 10,000
Averaged: no (1 run)
MNIST: 70,000

Accuracy:
94.55%

Loss history



Bonus: binary label optimizations

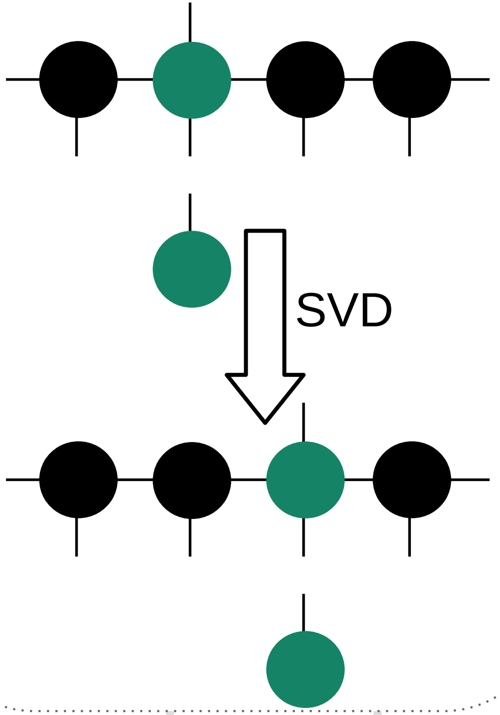
Label-link & argmax

(a, b)

↓ argmax(..)

0 if a > b else 1

Move label link



One-dimensional label link & rounding

(a)

↓ round(..)

0 if a < 0.5 else 1

SVD no longer needed
(no Krylov space needed)

Full and native support for half precision data type

Bonus: binary label runtimes

	float64	float32	float16
One-dimensional label link & rounding … with SVD	Accuracy: 99.75% T = 7.6s	Accuracy: 99.75% T = 6.9s	Accuracy: 99.7% T = 7.8s
One-dimensional label link & rounding … without SVD	Accuracy: 99.55% T = 4.4s	Accuracy: 99.55% T = 4.2s	Accuracy: 99.4% T = 4.3s

Memory-friendly

Conclusion and outlook

Quantum-inspired machine learning for supervised ML tasks

Hyper-parameters from quantum many-body physics

Option for GPU support + option to switch to torch

Also integrate jax and tensorflow as in DJ et al., arXiv 2409.03818

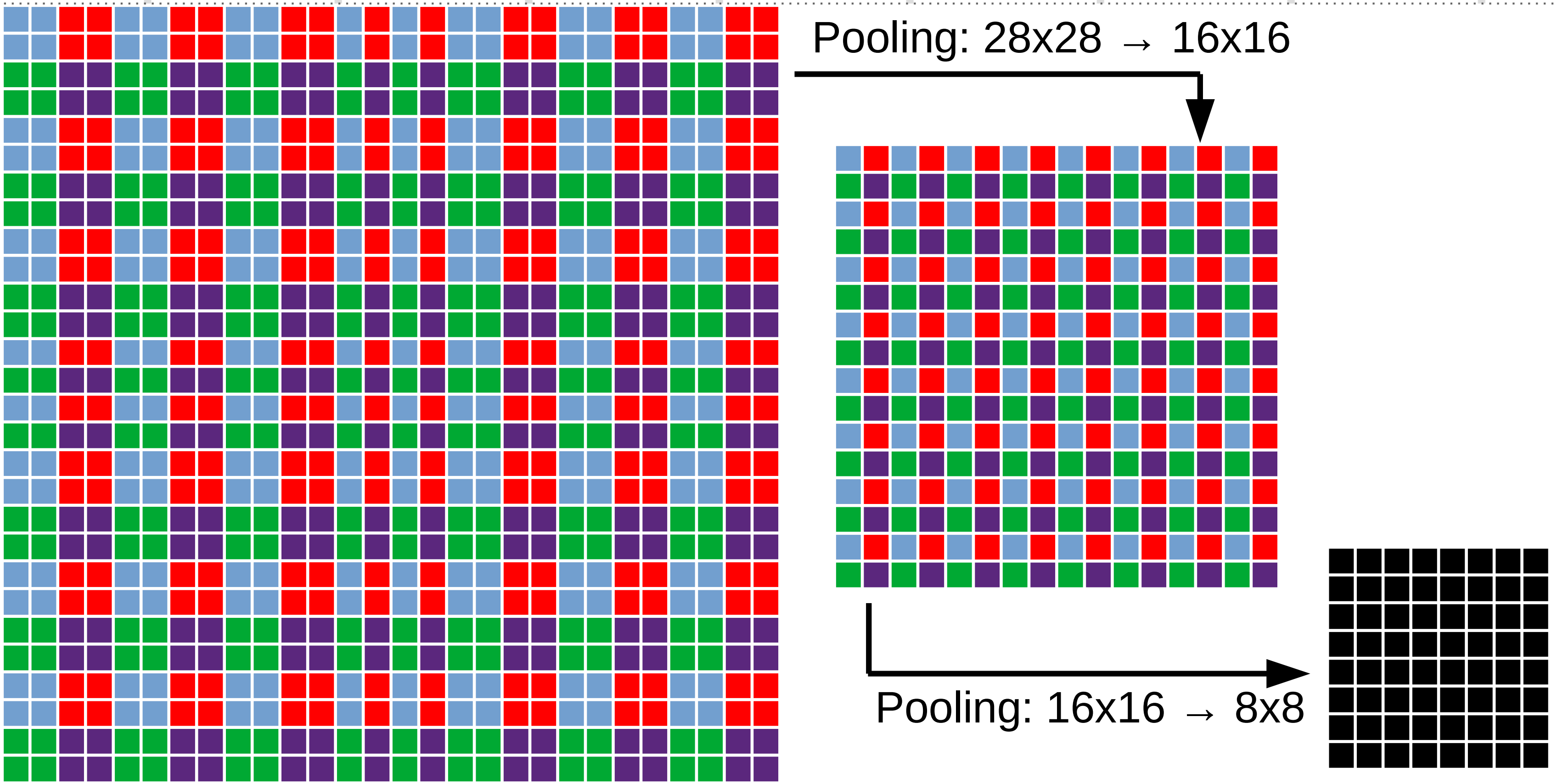
Explore high-energy data and follow path of FPGA triggers

See L. Borella, Alberto Coppi, et al. arxiv:2409.16075

SPONSORED BY THE Federal Ministry of Education and Research

Backup slides

Pooling layer



Modularity of the library

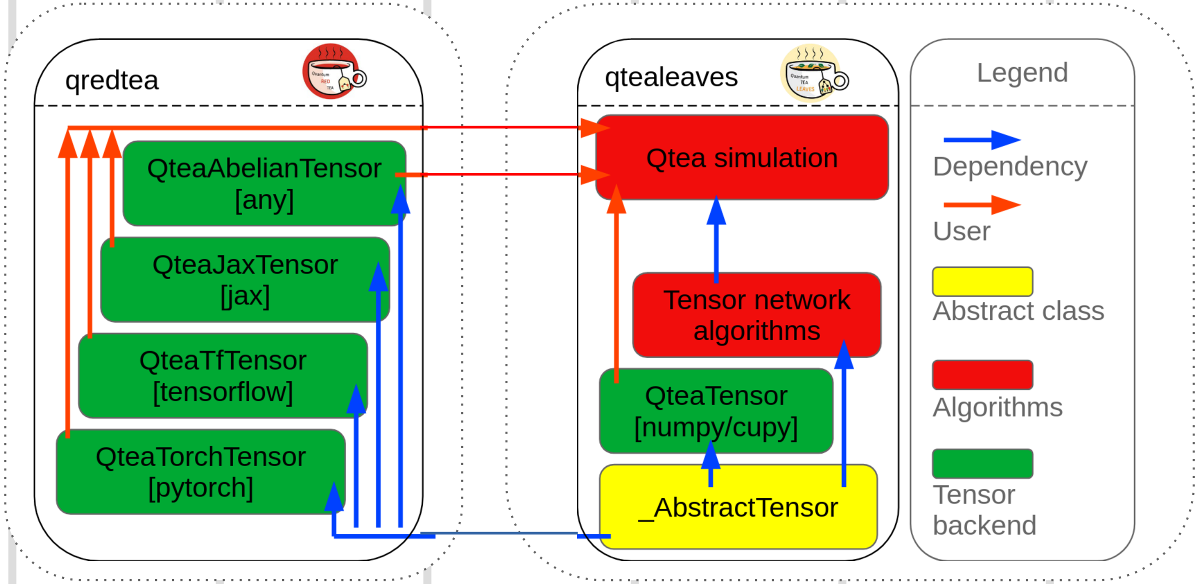
Why do you need another tea flavor?

How to integrate new technology in a library?

Can we use GPUs? … no, because ...

Can we use jax? … no, ...

Major update qtealeaves v1.0.0+



MPS (two-tensor update) versus TTN (single-tensor)