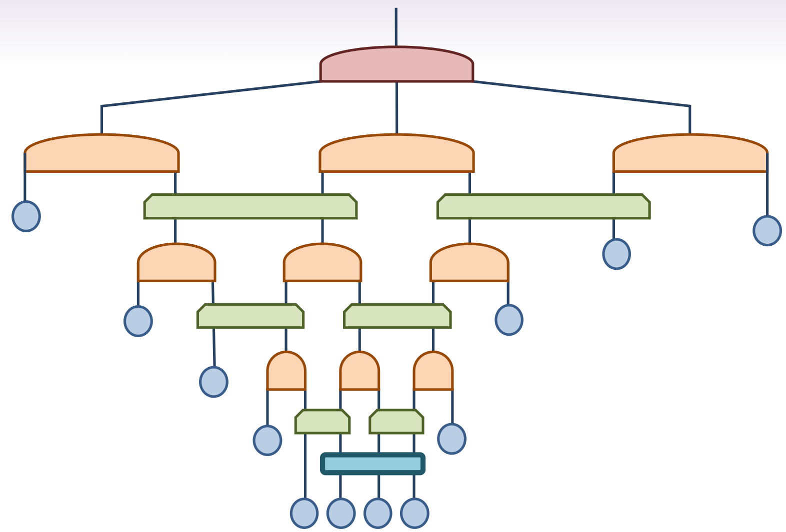
Entangle This IV: Chaos, Order and Qubits
Madrid, Sept 2019

Efficient use of tensor networks for tasks in supervised learning



Glen Evenbly

Georgia Tech

Number-State Preserving Tensor Networks as Classifiers for Supervised Learning
G.E., arXiv:1905.06352 (2019)

Before we begin… some advertisements…

Faculty Position - Tenured/Tenure Track Atomic, Molecular and Optical Physics (Georgia Institute of Technology)
The School of Physics of the Georgia Institute of Technology invites applications for a faculty position in Atomic, Molecular and Optical (AMO) Physics and related areas in Quantum Information Sciences (QIS), beginning Fall 2020. Appointments at the Assistant, Associate or Full Professor level will be considered, depending on qualifications.

Postdoc Position – within my group (Georgia Institute of Technology)
I am looking for a postdoc to join my group sometime in 2020 for a 2 year term. Applicants should have interest and experience in either tensor networks or some area of quantum info.

Come be my research gremlin in the tensor network mines of Atlanta!

Before we begin… some more advertisements…

Research website: www.tensors.net

A website designed to help people get started with the practical aspects of implementing tensor network algorithms:

- has tutorials with code examples which detail the basic skills (i.e. contracting a network)
- has example codes of many tensor network algorithms (**Exact Diagonalization**, **DMRG**, **TEBD**, **MERA**, **boundary MERA**, **TRG**, **TNR**, **PEPS**)
- all codes are available in MATLAB, Python and Julia languages

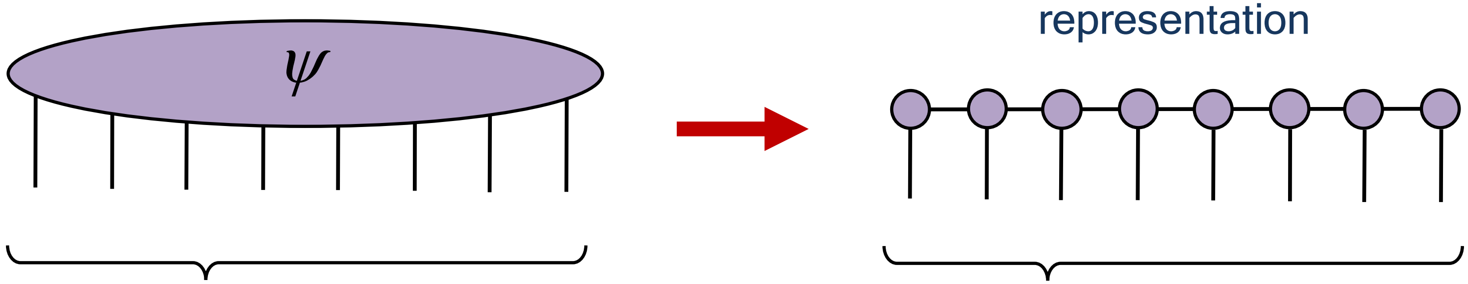
I have made an app called "TensorTrace" for designing and implementing tensor networks! Beta version available at:

www.tensortrace.com

Big Picture Overview: Tensor Networks

Many-body wavefunction
(describes state of lattice system)

Tensor network
representation



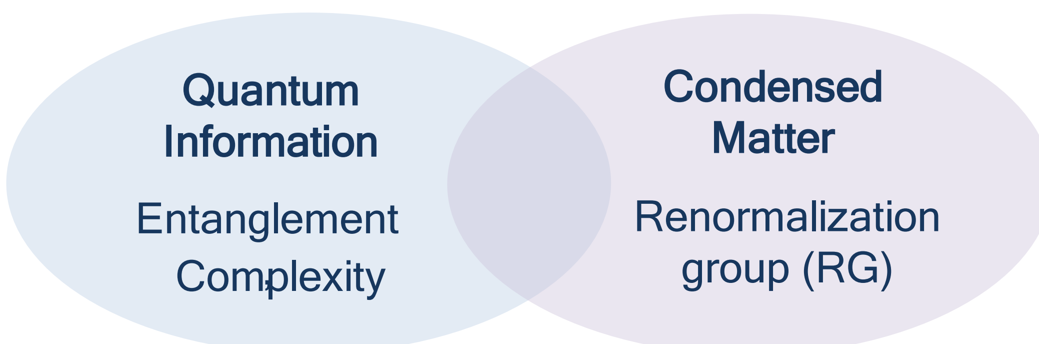
N - **body system**

N - **body system**

Complicated object!
exp(N) parameters

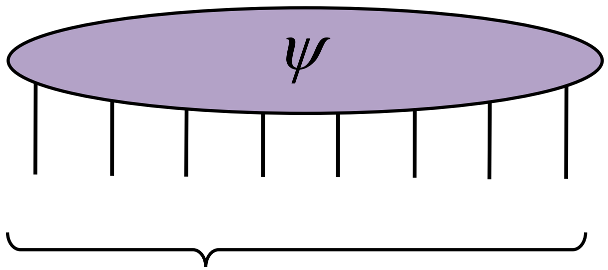
Product of simple objects!
poly(N) parameters

Tensor networks



Big Picture Overview: Tensor Networks

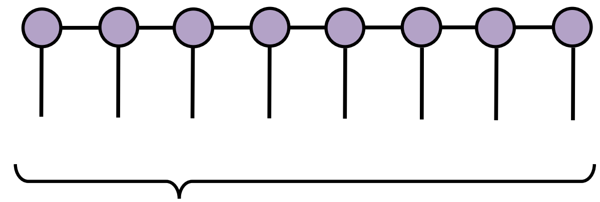
Many-body wavefunction
(describes state of lattice system)



N - **body system**

Complicated object!
exp(N) parameters

Tensor network
representation



N - **body system**

Product of simple objects!
poly(N) parameters

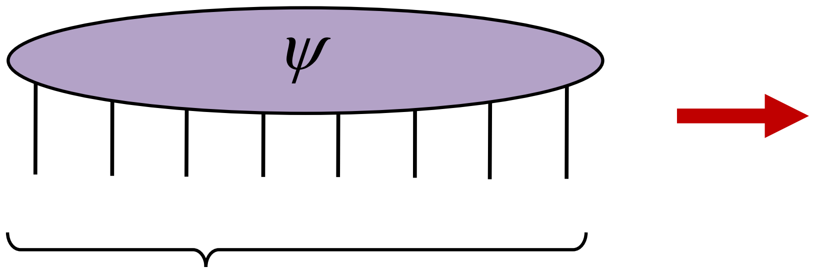
Practical goal: efficient numeric tools for classical simulation of quantum many-body systems

Theoretic goal: better understanding of many-body ground states

- Classification of phases of matter
- Which classes of quantum system can be efficiently simulated?
- Entanglement structure in many-body systems

Big Picture Overview: Tensor Networks

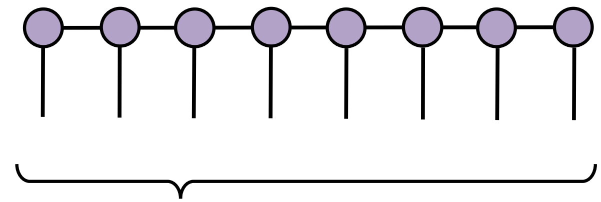
Many-body wavefunction
(describes state of lattice system)



N - **body system**

Complicated object!
exp(N) parameters

Tensor network
representation



N - **body system**

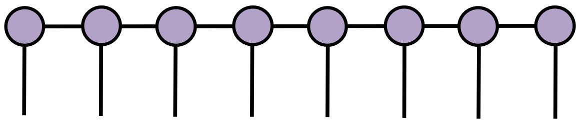
Product of simple objects!
poly(N) parameters

A tensor network is a compressed representation of some correlated data

Compression based on
tensor decompositions

Coefficients of a many-
body wavefunction

Big Picture Overview: Tensor Networks

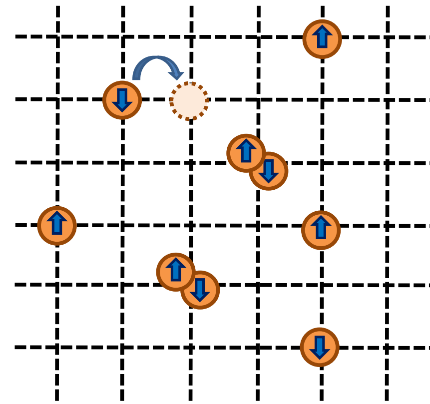


→ Study of quantum many-body systems

Tensor Network

||

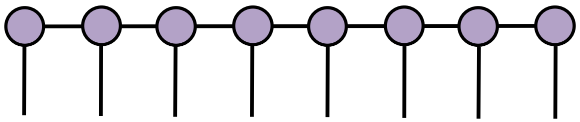
Compressed representation of some correlated data



Tensor network formalism has a wide variety of uses!

(e.g. interacting fermions on a lattice)

Big Picture Overview: Tensor Networks



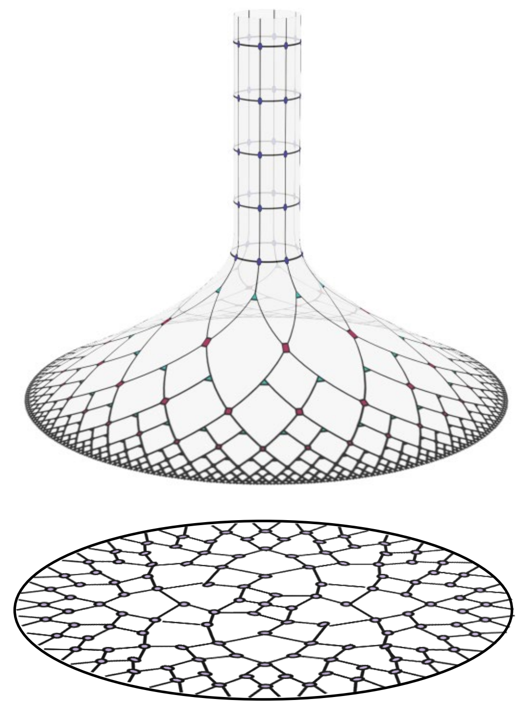
Study of quantum many-body systems

Holography: duality between semi-classical gravity and conformal field theories

Tensor Network

||

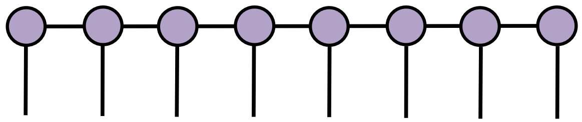
Compressed representation of some correlated data



Tensor network formalism has a wide variety of uses!

(network as a discretization of space-time)

Big Picture Overview: Tensor Networks



Tensor Network

||

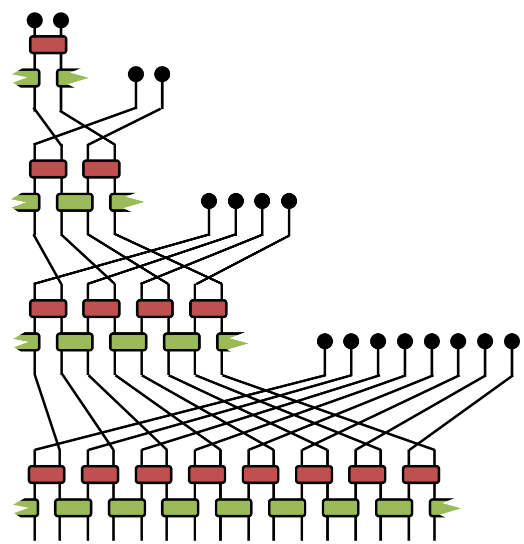
Compressed representation of some correlated data

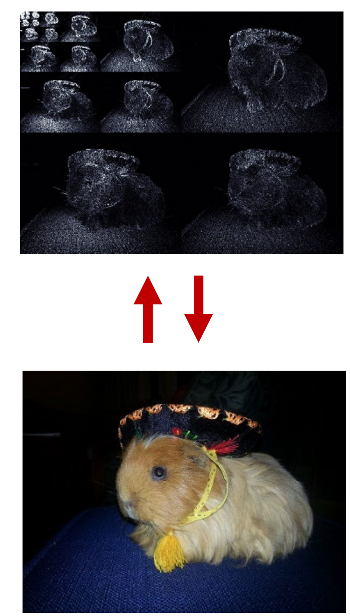
Study of quantum many-body systems

Holography: duality between semi-classical gravity and conformal field theories

Data compression: multi-resolution analysis and wavelets

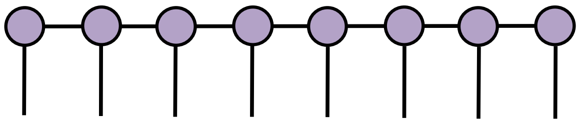
Tensor network formalism has a wide variety of uses!





(e.g. image compression)

Big Picture Overview: Tensor Networks



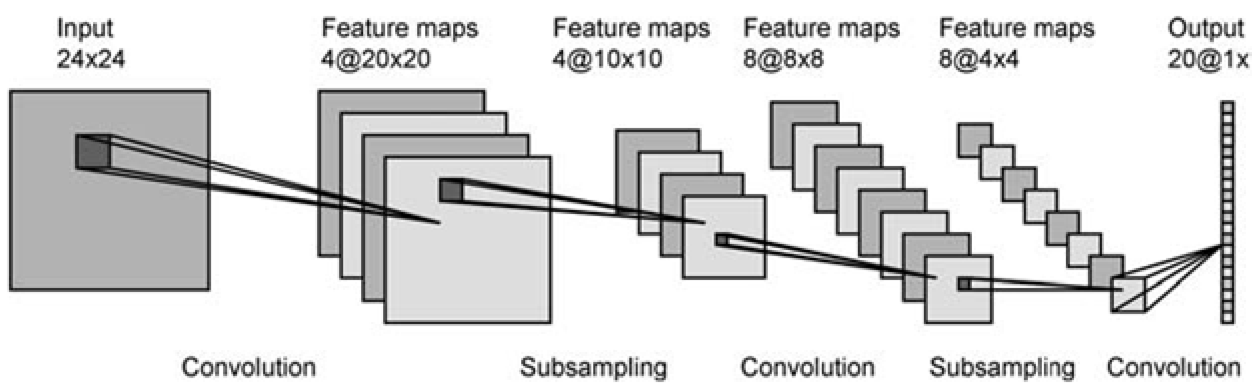
Tensor Network

||

Compressed representation of some correlated data

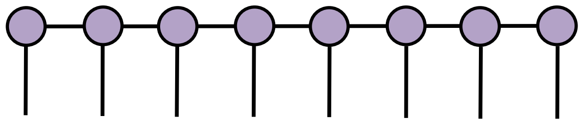
- Study of quantum many-body systems
- Holography: duality between semi-classical gravity and conformal field theories
- Data compression: multi-resolution analysis and wavelets
- Machine learning: neural networks

Tensor network formalism has a wide variety of uses!



(e.g. convolutional neural network)

Big Picture Overview: Tensor Networks



Tensor Network

||

Compressed representation of some correlated data

Tensor network formalism has a wide variety of uses!

- Study of quantum many-body systems
- Holography: duality between semi-classical gravity and conformal field theories
- Data compression: multi-resolution analysis and wavelets
- Machine learning: neural networks
- Error correcting codes, e.g. polar codes
- Big data analytics

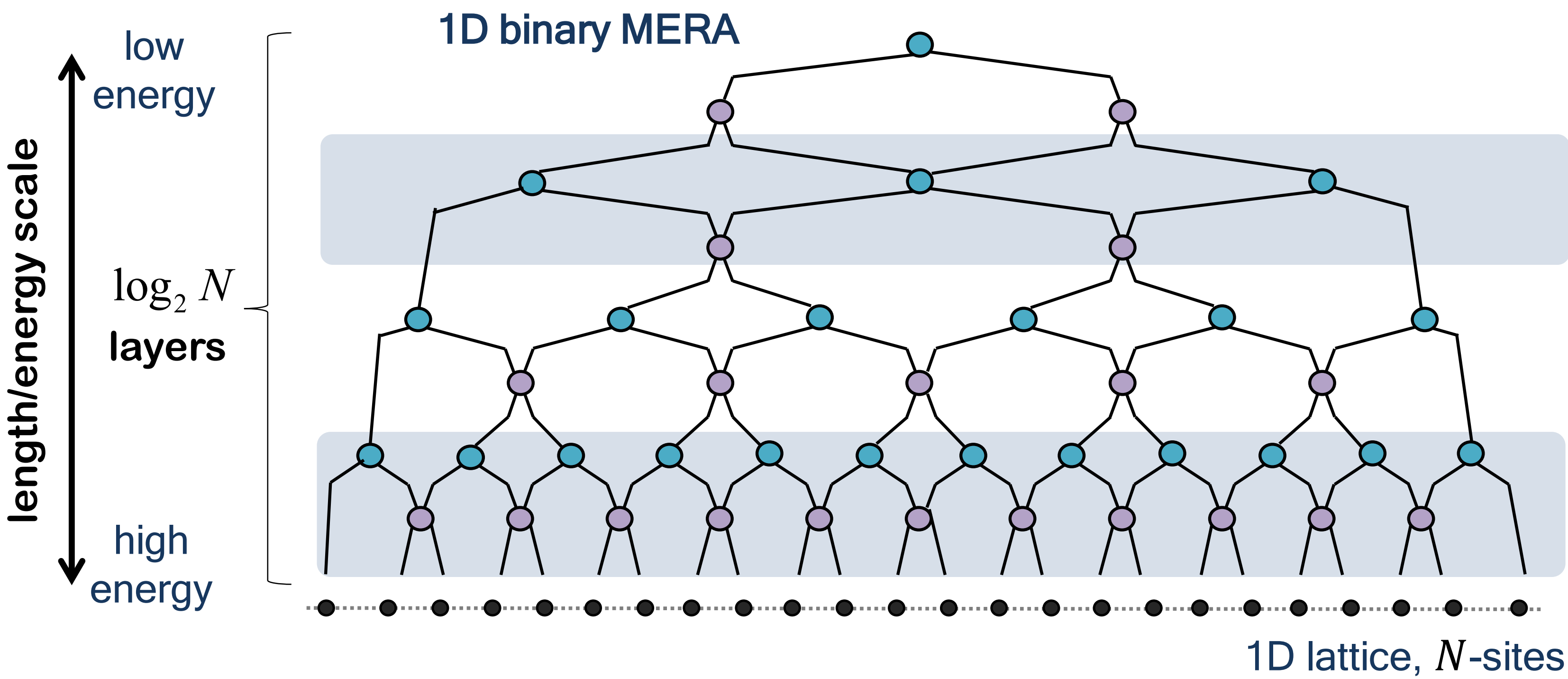
+ many more!

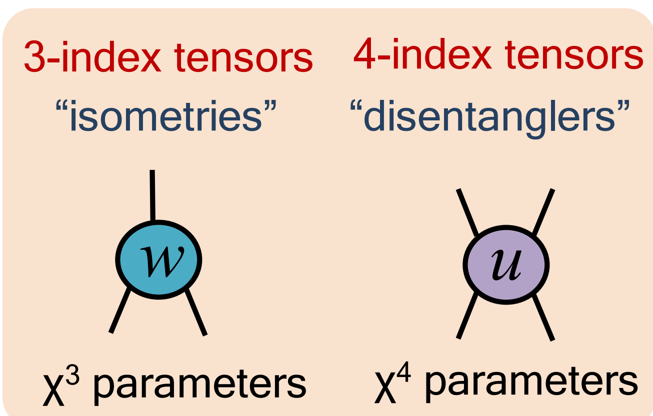
Ideas developed in the context of **entanglement** and efficient representation of quantum **wavefunctions** are useful in many areas outside of physics

G.E., Steven. R. White, **Phys. Rev. Lett 116.** 140403 (2016)
G.E., Steven. R. White, **Phys. Rev. A 97,** 052314 (2018)
J. Haegeman, B. Swingle, M. Walter, J. Cotlet, **G.E.**, V. Scholz, **Phys. Rev. X 8,** 111003 (2018)

Multi-scale entanglement renormalization ansatz (MERA)

G. Vidal, **PRL 101, 110501** (2008)





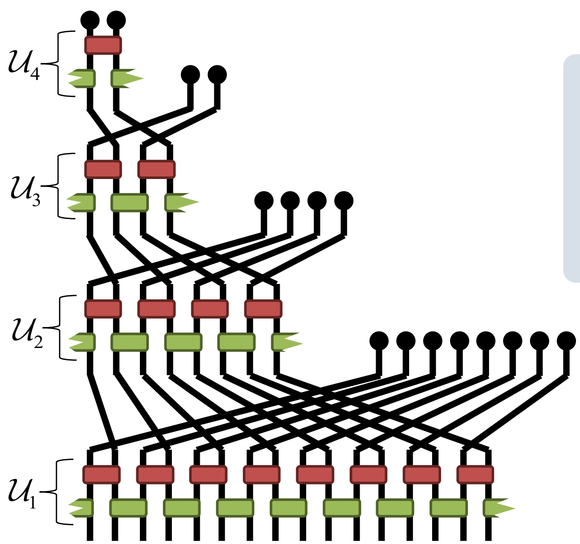
Multi-scale decomposition of the wavefunction:

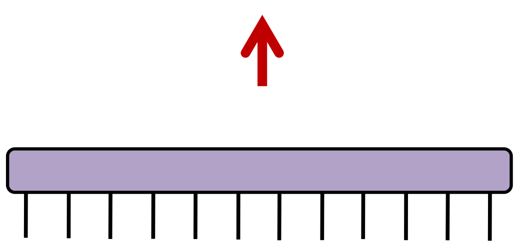
- lower layers encode short-ranged (high energy) properties of the state
- higher layers encode long-ranged (low energy) properties of the state

Wavelets and tensor networks

MERA : multi-scale decomposition of many-body wavefunction

Long scale info ⟷ **Tensor Network** ⟷ Short scale info



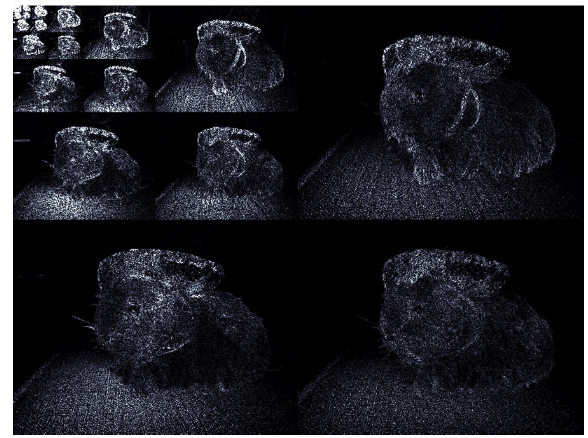


Many-body wavefunction

Proposal by Steve White: wavelets and MERA are connected ⟷

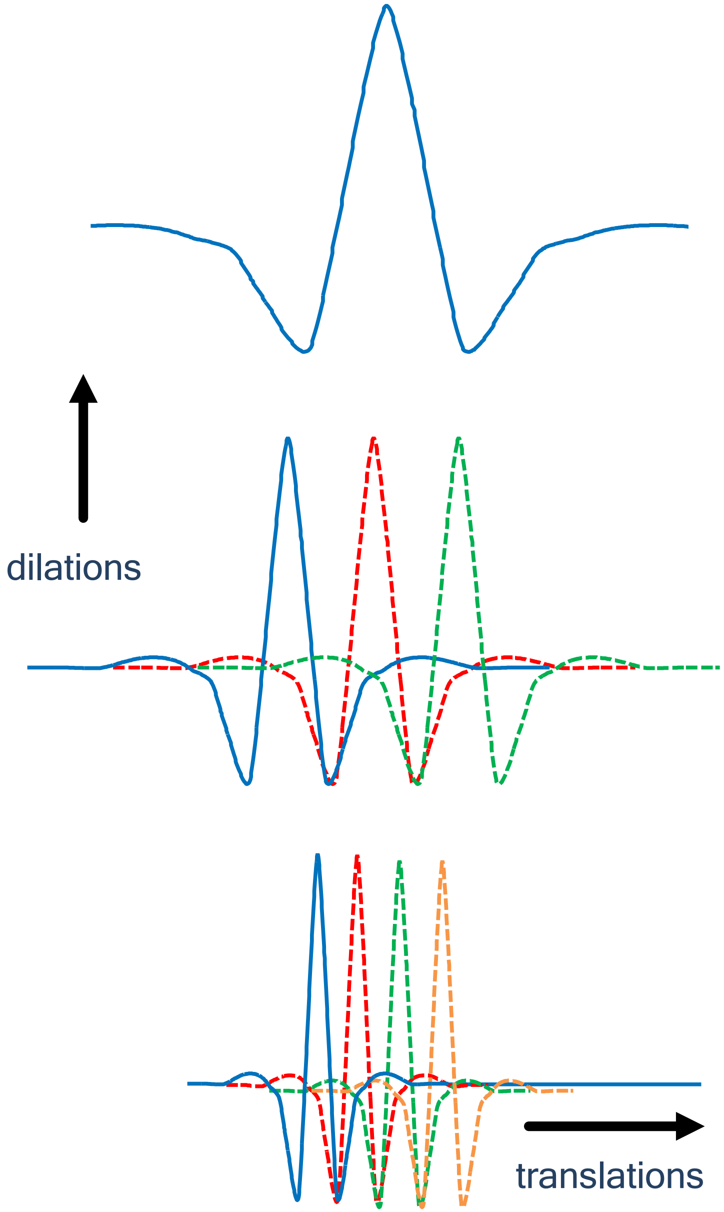
Discrete wavelet transform: multi-resolution analysis of classical data

Long scale info ⟷ **Wavelet basis** ⟷ Short scale info



Image

Introduction to Wavelets



Wavelet basis

- basis consists of translations and dilations of a wavelet function
- is a multi-resolution analysis (MRA)

Wavelets are a good compromise between real-space and Fourier-space representations

- compact in real-space and in frequency-space
- developed by math and signal processing communities in late 80's
- applications in signal and image processing, data compression (e.g. JPEG2000 image format)

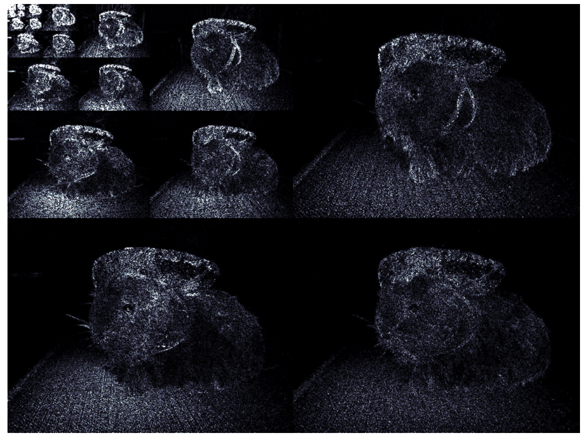
Image compression with wavelets

Long scale info

Wavelet basis

Short scale info

image



transform to wavelet basis

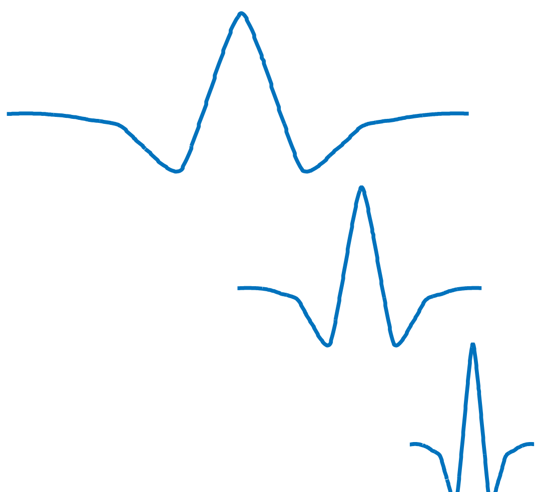
truncate (keep only largest 2% of coefficients)

inverse transform

compressed image

peak signal to noise:

PSNR: 37.0 dB

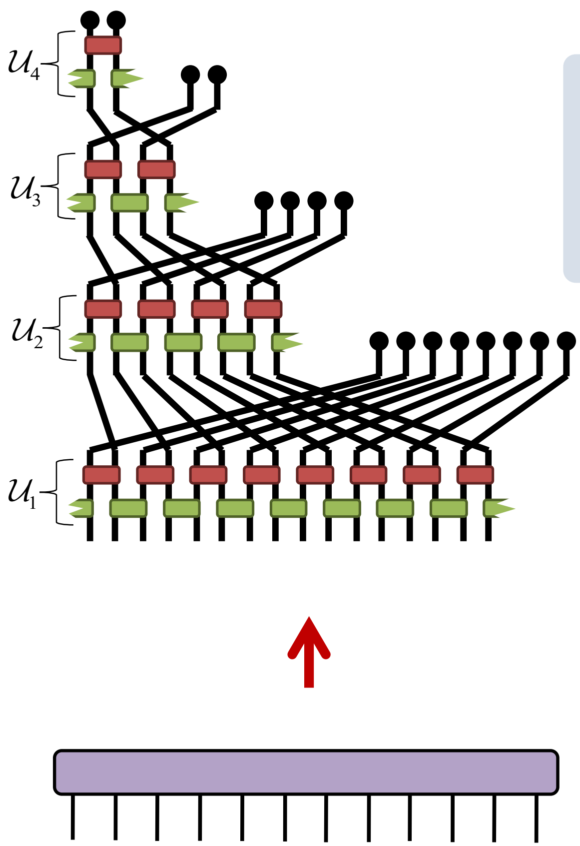


This is the key part of JPEG2000 format, and many other standards for image, audio and video compression

Wavelets and tensor networks

MERA : multi-scale decomposition of many-body wavefunction

Long scale info | **Tensor Network** ⟷ | Short scale info



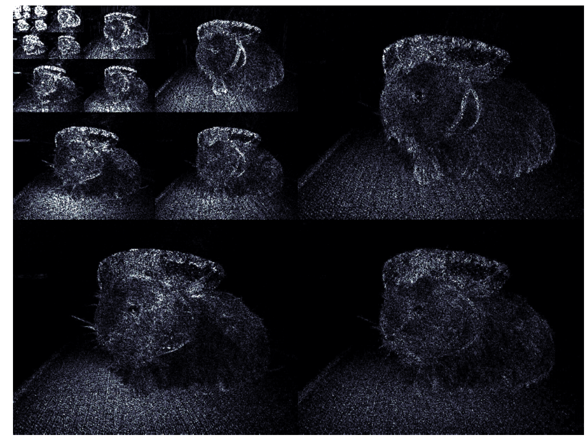
Many-body wavefunction

Many concrete connections can be made ⟷

studying these connections proved very useful!

Discrete wavelet transform: multi-resolution analysis of classical data

Long scale info | **Wavelet basis** ⟷ | Short scale info

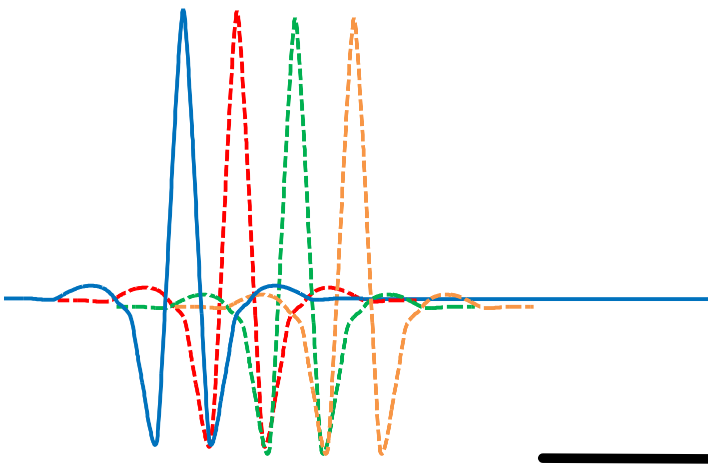


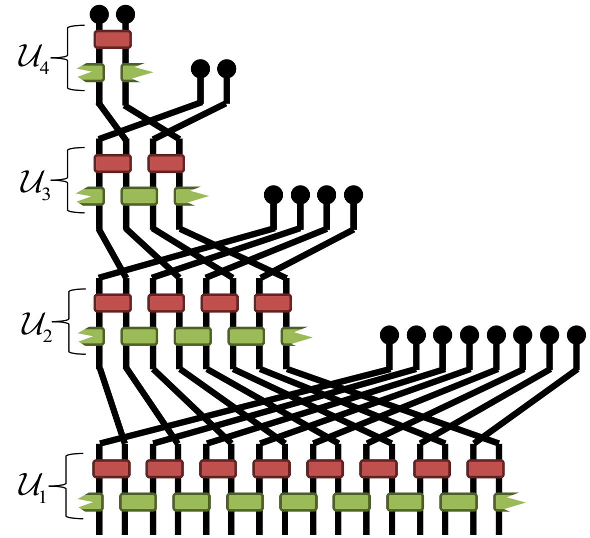
Image

Wavelets and tensor networks

Discrete wavelet transform: multi-resolution analysis of classical data

MERA : multi-scale decomposition of many-body wavefunction





First analytic MERA for critical states

G.E., Steven. R. White, **Phys. Rev. Lett 116.** 140403 (2016)

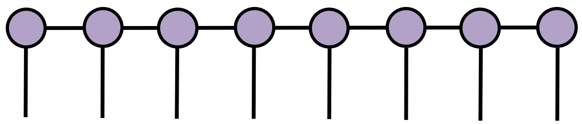
First rigorous error bounds for MERA accuracy

J. Haegeman, B. Swingle, M. Walter, J. Cotlet, **G.E.**, V. Scholz, **Phys. Rev. X 8,** 111003 (2018)

Ideas from tensor networks used to construct new and improved wavelet transformations

G.E., Steven. R. White, **Phys. Rev. A 97,** 052314 (2018)

Other Applications of Tensor Networks

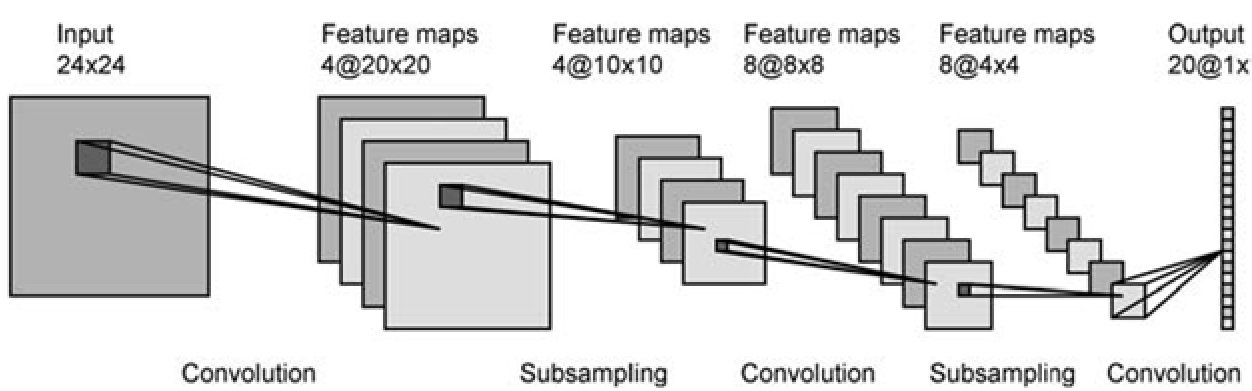


Tensor Network

||

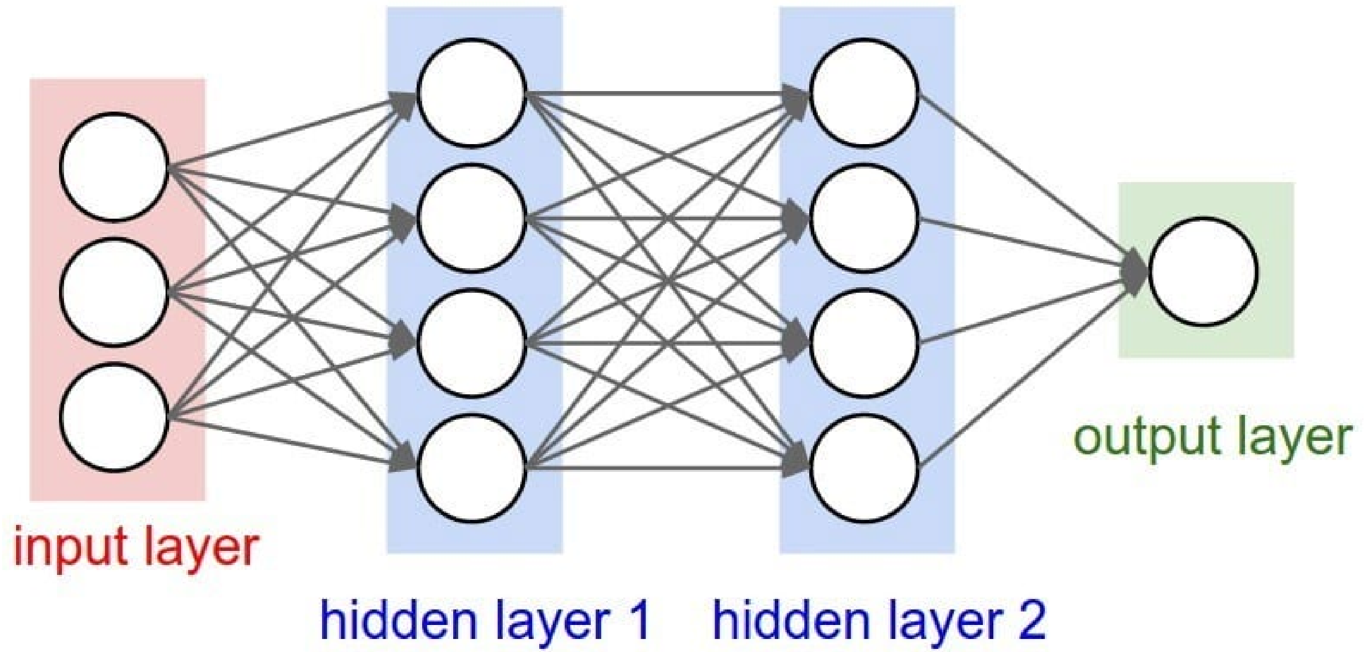
Compressed representation of some correlated data

- Study of quantum many-body systems
- **Holography**: duality between semi-classical gravity and conformal field theories
- **Data compression**: multi-resolution analysis and wavelets
- **Machine learning**: neural networks



Machine Learning and tensor networks

In recent times deep neural networks have been spectacularly successful



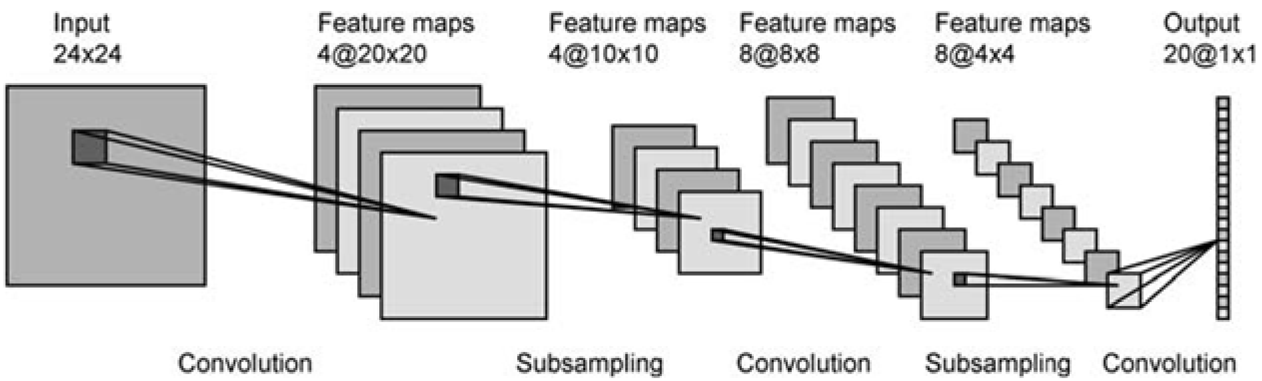
Superficially deep neural networks look very similar to tensor networks. Are there connections?

Can ideas from machine learning be used to improve simulation algorithms for quantum systems?

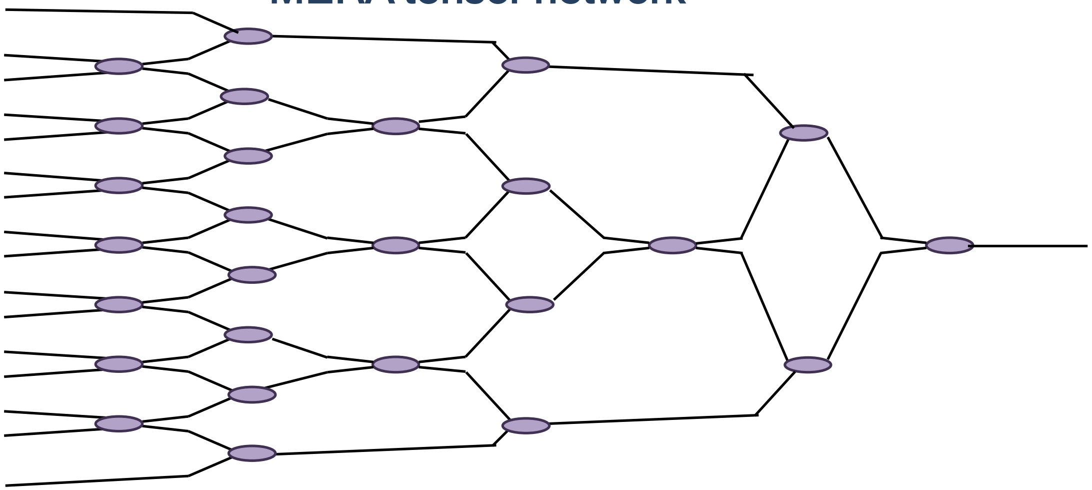
Can ideas from tensor networks be used to improve machine learning and neural networks?

Machine Learning and tensor networks

Convolutional neural network (CNN)



MERA tensor network



convolutional neural networks are structurally very similar to MERA tensor network!

Papers using tensor network machine learning

Expressivity & priors of TN based models

- *Levine et al., "**Deep Learning and Quantum Entanglement: Fundamental Connections with Implications to Network Design**" arxiv:1704.01552*
- *Cohen, Shashua, "**Inductive Bias of Deep Convolutional Networks through Pooling Geometry**" arxiv:1605.06743*
- *Cohen et al., "**On the Expressive Power of Deep Learning: A Tensor Analysis**" arxiv:1509.05009*

Generative Models

- *Han et al., "**Unsupervised Generative Modeling Using Matrix Product States**" arxiv:1709.01662*
- *Sharir et al., "**Tractable Generative Convolutional Arithmetic Circuits**" arxiv:1610.04167*

Supervised Learning

- *Novikov et al., "**Expressive power of recurrent neural networks**", arxiv:1711.00811*
- *Liu et al., "**Machine Learning by Two-Dimensional Hierarchical Tensor Networks: A Quantum Information Theoretic Perspective on Deep Architectures**", arxiv:1710.04833*
- *Stoudenmire, Schwab, "**Supervised Learning with Quantum-Inspired Tensor Networks**", arxiv:1605.05775*
- *Novikov et al., "**Exponential Machines**", arxiv: 1605.03795*

Related uses of tensor networks

Compressing weights of neural nets (& other models)

Yu et al., Advances in Neural Information Processing (2017), arxiv:1711.00073

Izmailov et al., arxiv:1710.07324 (2017)

Yang et al., arxiv:1707.01786 (2017)

Garipov et al., arxiv:1611.03214 (2016)

Novikov et al., Advances in Neural Information Processing (2015) (arxiv:1509.06569)

Large scale linear algebra (PCA/SVD)

Lee, Cichocki, arxiv: 1410.6895 (2014)

Feature extraction & tensor completion

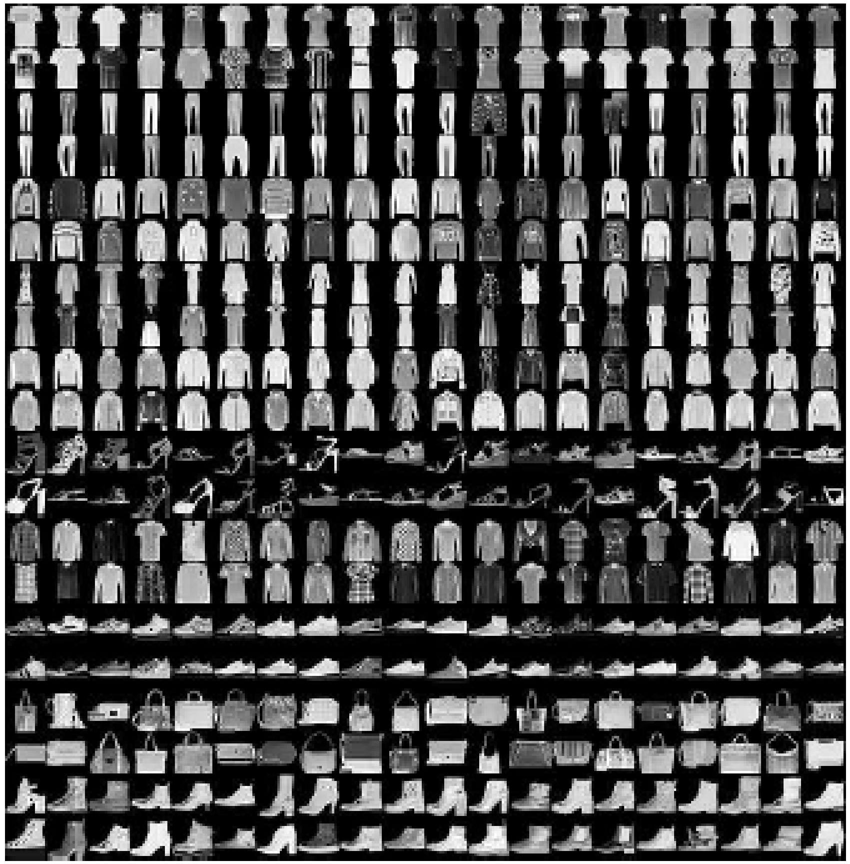
Bengua et al., arxiv:1606.01500, arxiv:1607.03967, arxiv:1609.04541 (2016)

Phien et al., arxiv:1601.01083 (2016)

Bengua et al., IEEE Congress on Big Data (2015)

Machine learning test problem: Fashion MNIST

Label	Description	Examples
0	T-Shirt/Top	
1	Trouser	
2	Pullover	
3	Dress	
4	Coat	
5	Sandals	
6	Shirt	
7	Sneaker	
8	Bag	
9	Ankle boots	



Test Problem:
Fashion MNIST database

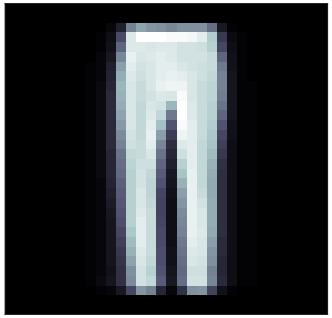
- 10 classes of clothing
- 60,000 training images
- 10,000 test images

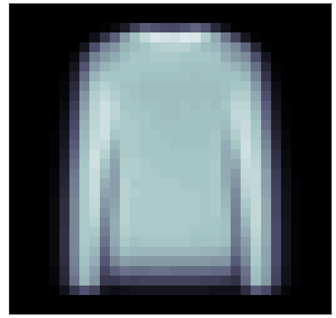
Goal: train a program to properly classify the test images

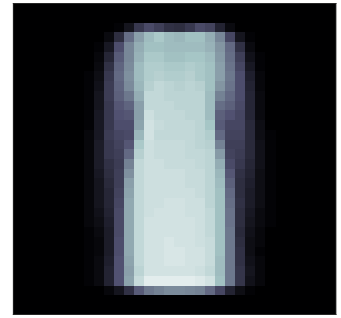
Simplest approach: compare test images again the "average" of all training images in a given category (related to principle component analysis)

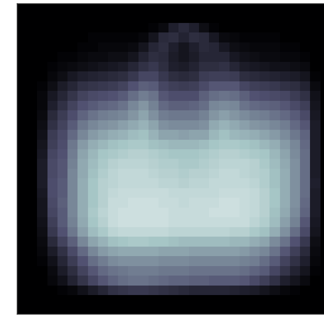
Machine learning test problem: Fashion MNIST

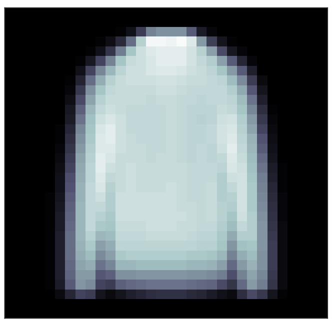

T-shirt/top

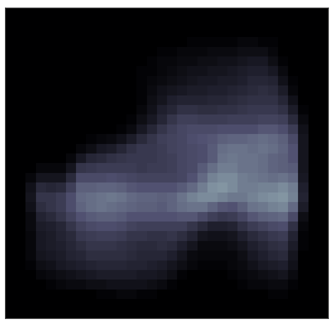

Trousers

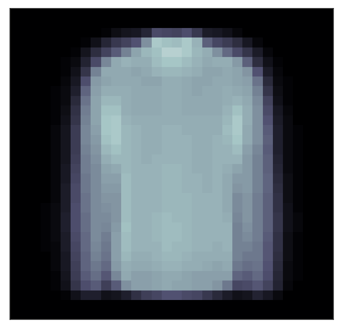

Pullover

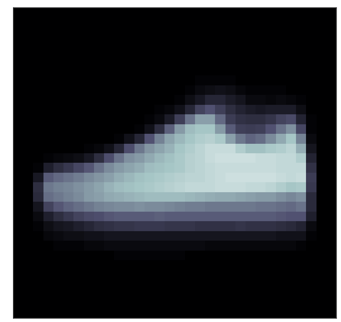

Dress

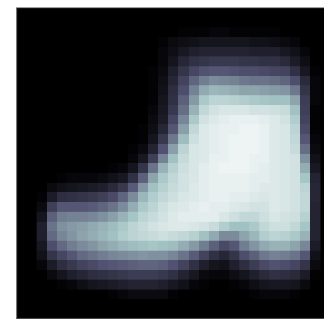

Bag


Coat


Sandal


Shirt


Sneaker


Ankle boot

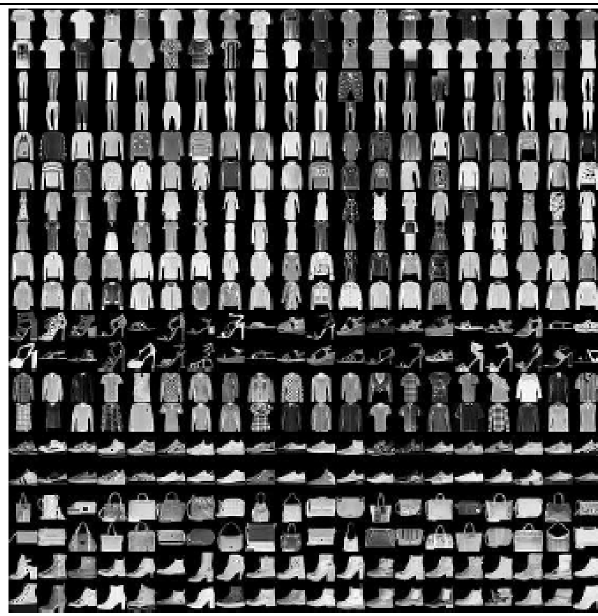
Simplest approach: compare test images again the "average" of all training images in a given category (related to principle component analysis)

- gives 66% correct

More advanced machine learning approaches?

Convolutional Neural Networks

Label	Description	Examples
0	T-Shirt/Top	
1	Trouser	
2	Pullover	
3	Dress	
4	Coat	
5	Sandals	
6	Shirt	
7	Sneaker	
8	Bag	
9	Ankle boots	



Fashion MNIST test problem:
some benchmarks (no preprocessing)

- XGBoost (89.8%)
- AlexNet (89.9%)
- two-layer CNN trained with Keras (87.6%)
- GoogLeNet (93.7%)

Can tensor network methods be applied to this problem (and other machine learning problems)?

Original study by Miles Stoudenmire

E. M. Stoudenmire, Quantum Sci. and Technol. 3, 034003 (2018)

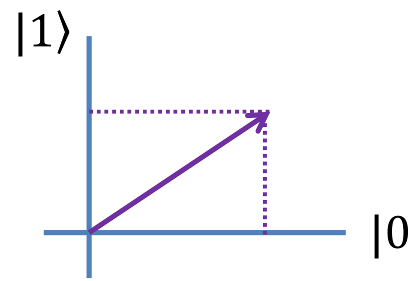
Reproduced using a similar method by me!

Tensor networks for image classification

Greyscale pixel

$$|\psi_i\rangle = \cos\left(\frac{\pi\alpha_i}{2}\right)|0\rangle + \sin\left(\frac{\pi\alpha_i}{2}\right)|1\rangle$$

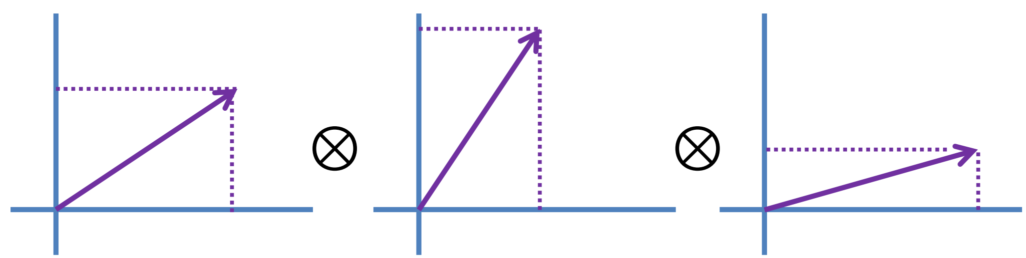
Qubit state (vector in d=2 dims)



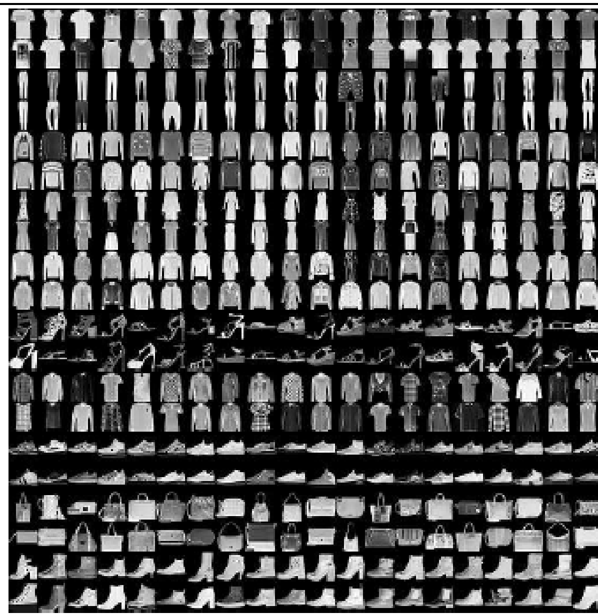
Greyscale image (N pixels)

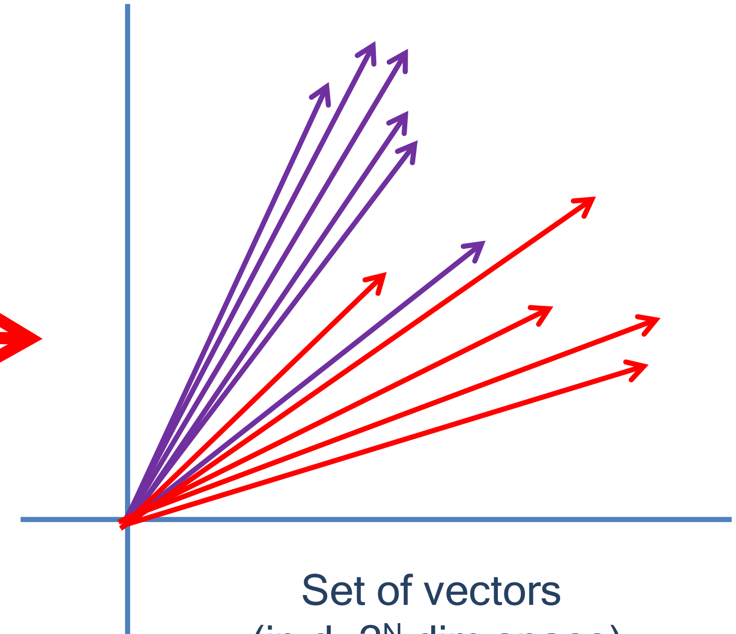
$$|\Psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes |\psi_3\rangle \otimes |\psi_4\rangle \otimes \cdots$$

Product state (vector in $d=2^N$ dims)



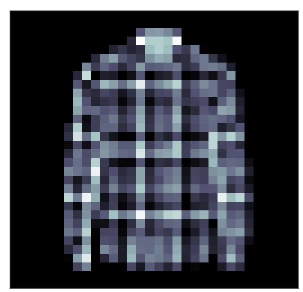
Tensor networks for image classification

Label	Description	Examples
0	T-Shirt/Top	
1	Trouser	
2	Pullover	
3	Dress	
4	Coat	
5	Sandals	
6	Shirt	
7	Sneaker	
8	Bag	
9	Ankle boots	



Set of vectors
(in $d=2^N$ dim space)

Each test image is a product state on a 2D lattice of qubits:

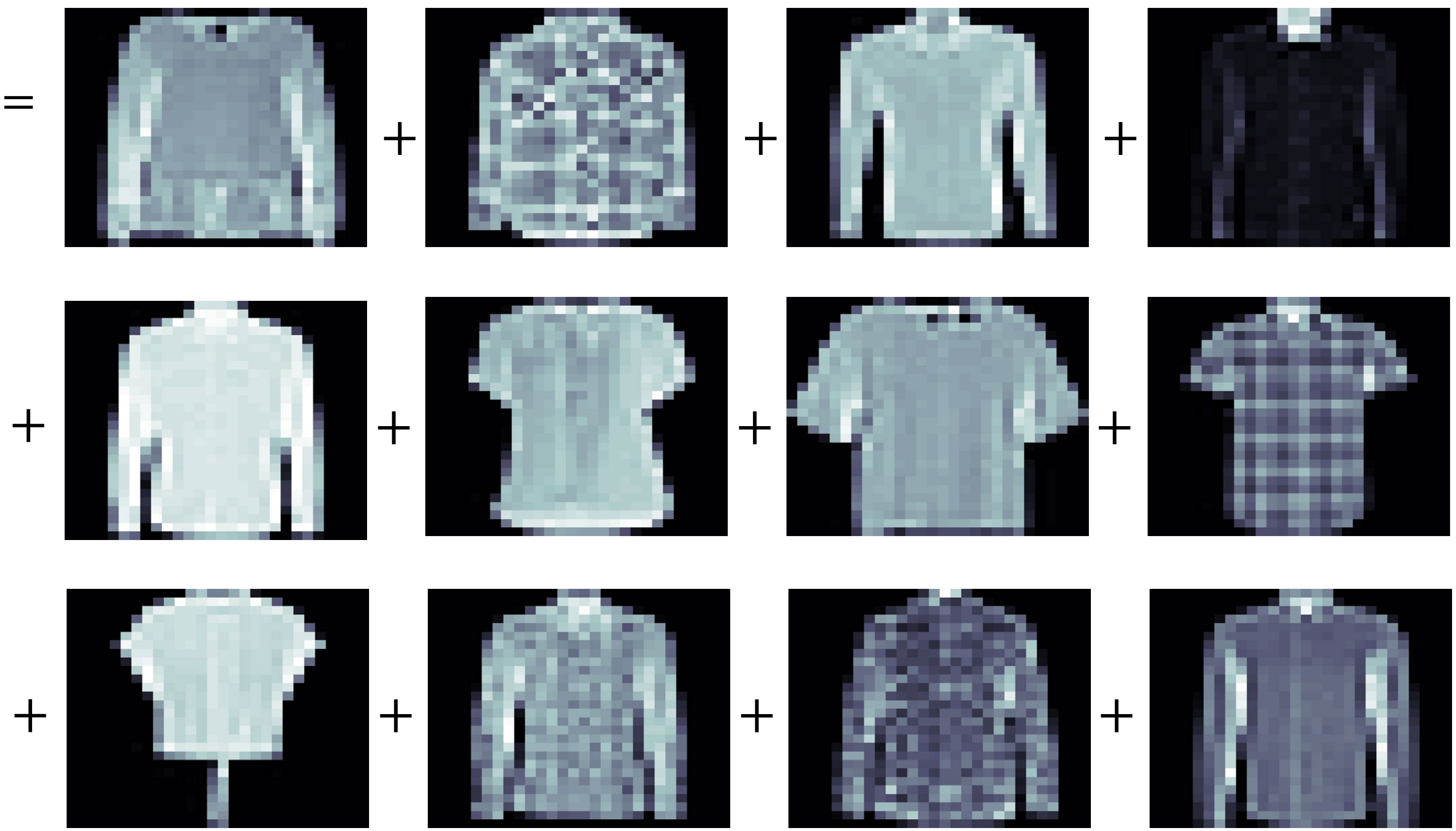
$|\Psi_1\rangle =$  $|\Psi_2\rangle =$  $|\Psi_3\rangle =$ 

Can we form the (entangled) superposition
of all images from a given class?

Tensor networks for image classification

$|\Psi_{\text{shirt}}\rangle =$ superposition of all pixel images of shirts

$|\Psi_{\text{shirt}}\rangle =$



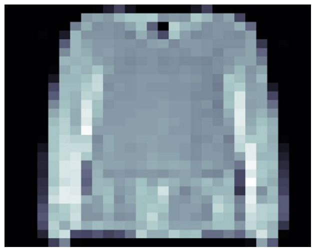
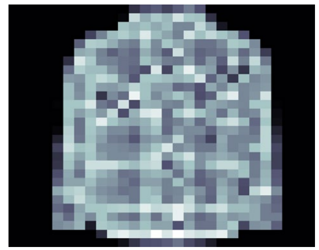
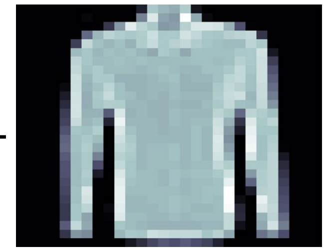
$+ \cdots$

Is this state sufficiently low in entanglement that it can be approximated as a tensor network?

Can tensor network algorithms be adapted to efficiently construct this approximation?

Tensor networks for image classification

$|\Psi_{\text{shirt}}\rangle =$ superposition of all pixel images of shirts

$|\Psi_{\text{shirt}}\rangle =$  +  +  + 

Encode the training data as a **2D tree tensor network**

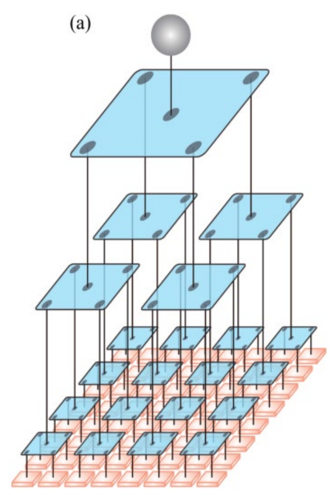


Image classification problem

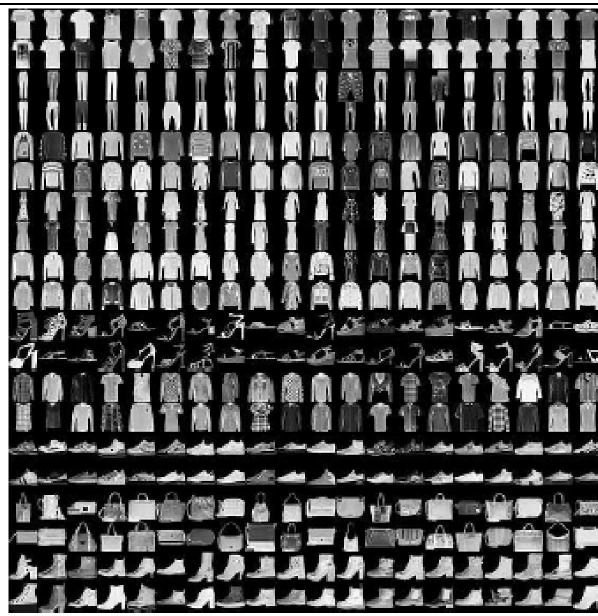
↓

Quantum many-body problem

Is this state sufficiently low in entanglement that it can be approximated as a tensor network?

Can tensor network algorithms be adapted to efficiently construct this approximation?

Tensor networks for image classification

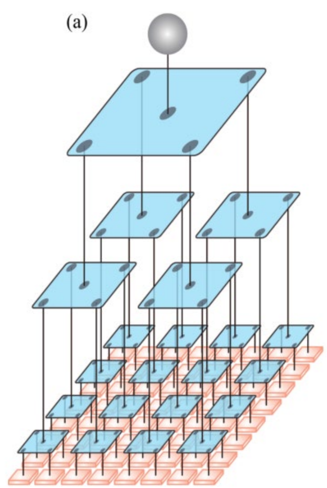
Label	Description	Examples
0	T-Shirt/Top	
1	Trouser	
2	Pullover	
3	Dress	
4	Coat	
5	Sandals	
6	Shirt	
7	Sneaker	
8	Bag	
9	Ankle boots	

Fashion MNIST test problem:
some benchmarks (no preprocessing)

- XGBoost (89.8%)
- AlexNet (89.9%)
- two-layer CNN trained with Keras (87.6%)
- GoogLeNet (93.7%)

Tree tensor network

(4-to-1 blocking scheme)

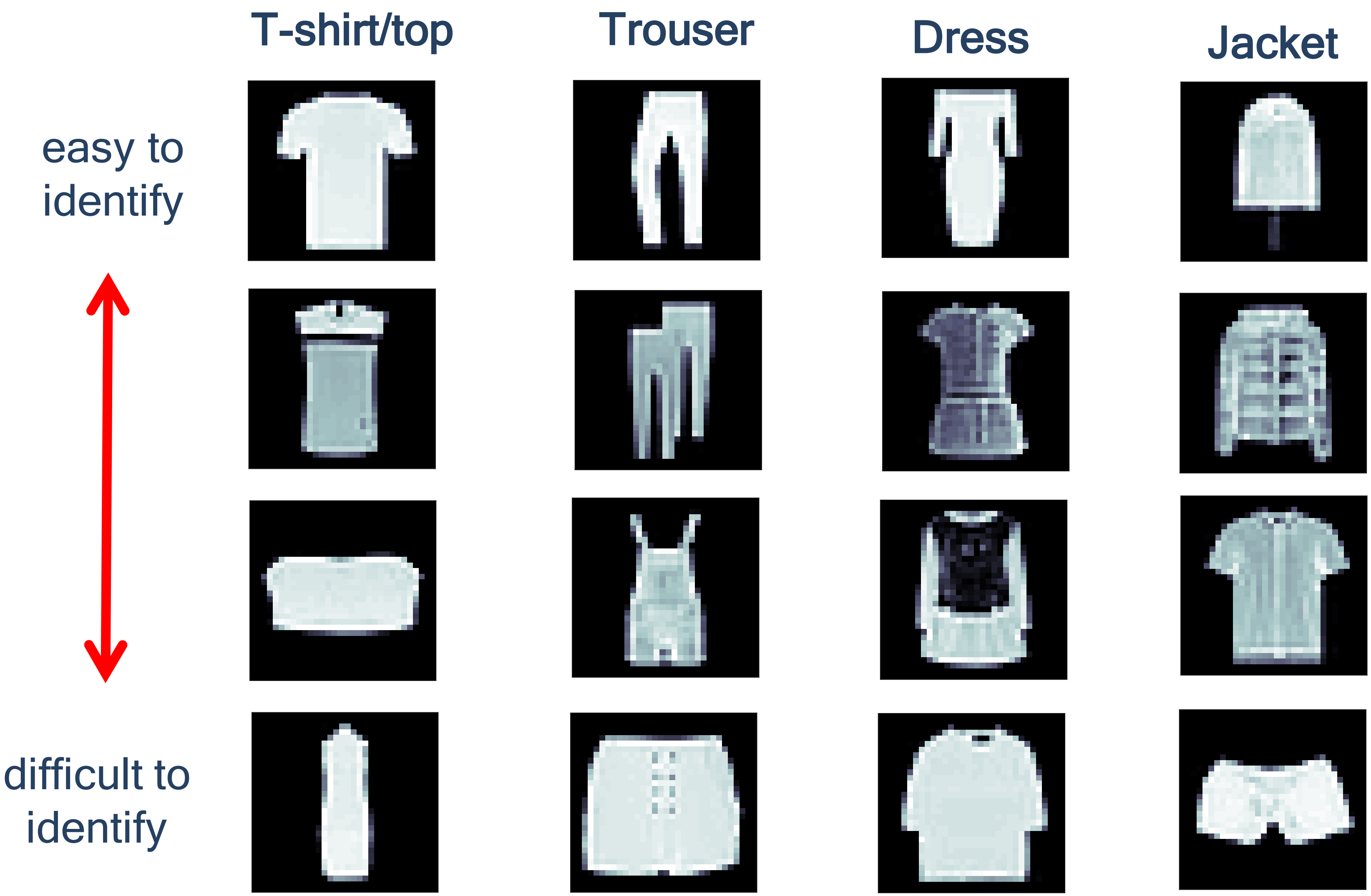


bond dim χ = 64

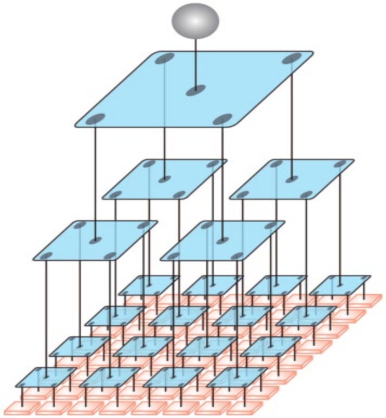
training time ~ 60min (laptop)

- gives 89.5% correct

Tensor networks for image classification



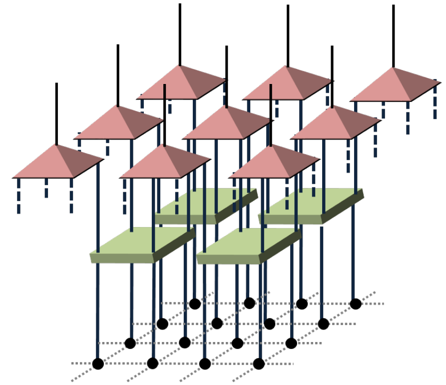
Tensor networks for image classification



Tree tensor network (TTN) + standard tensor network optimization strategies

↓

Performs quite well in benchmark image classification problem



Can we instead use a multi-scale entanglement renormalization ansatz (MERA)?

Why?

- MERA are expected to greatly outperform a TTN for a 2D problem
- MERA are the natural analogue to convolutional neural networks (CNNs)

There are some problems with trying to use a MERA!

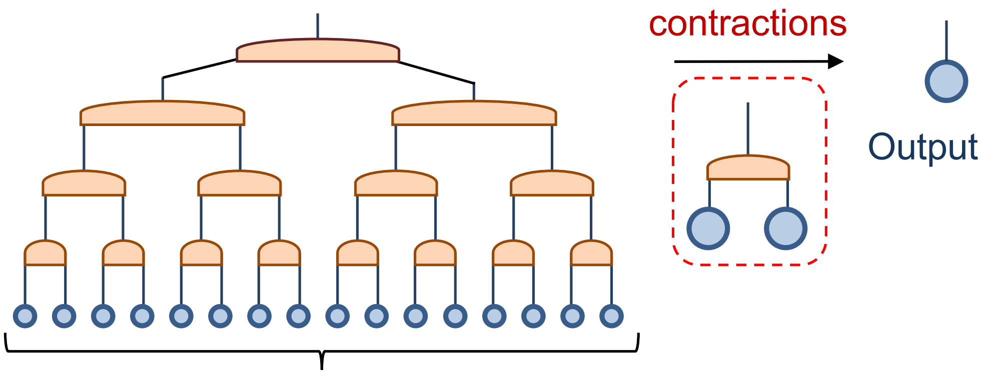
(1) Computational problem

(2) Conceptual problem

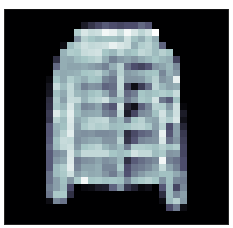
Tensor networks for machine learning

(1) Computational problem

Classification with a (trained) TTN:



Input sample
(product state)



Which category does this belong in?

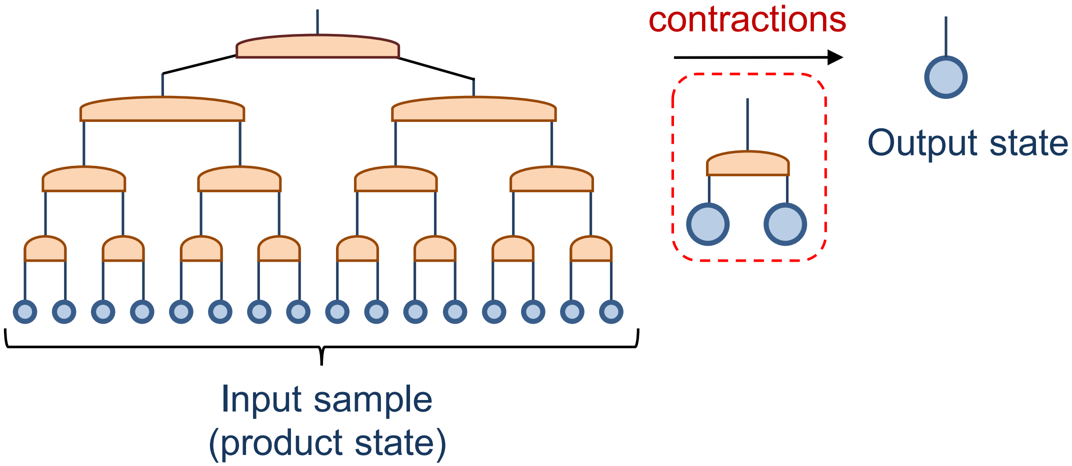
Contraction of TTN with a product state is efficient!

- classification of samples is efficient
- training (or optimization) of TTN can be done efficiently

Tensor networks for machine learning

(1) Computational problem

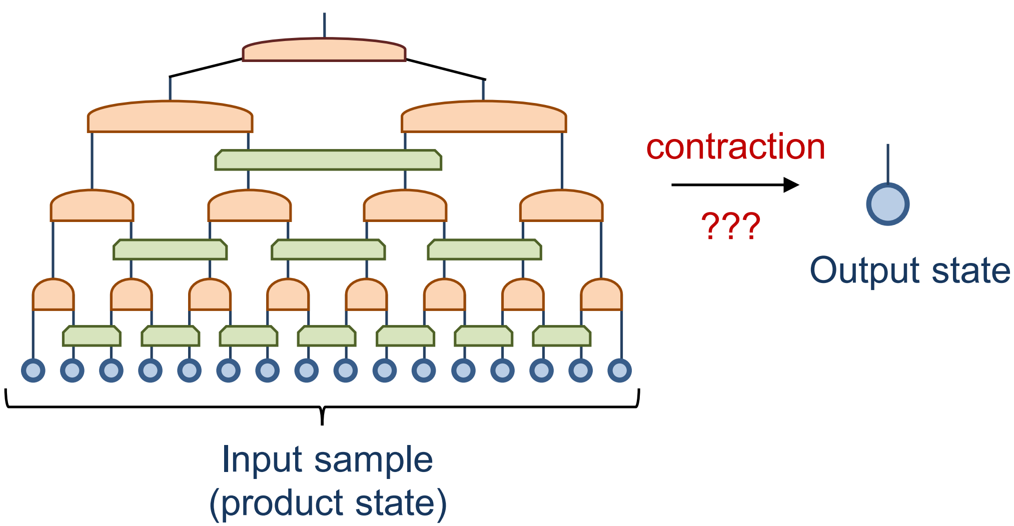
Classification with a (trained) TTN:



Contraction of TTN with a product state is efficient!

- classification of samples is efficient
- training (or optimization) of TTN can be done efficiently

Classification with a (trained) MERA?

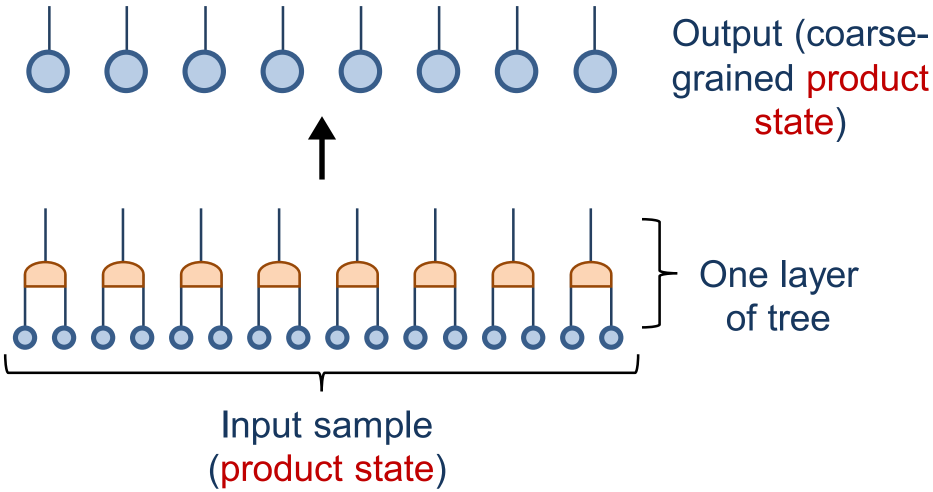


Contraction is of MERA with a product state is not efficient!

- classification of samples is not efficient
- training (or optimization) of MERA cannot be done efficiently for large problems

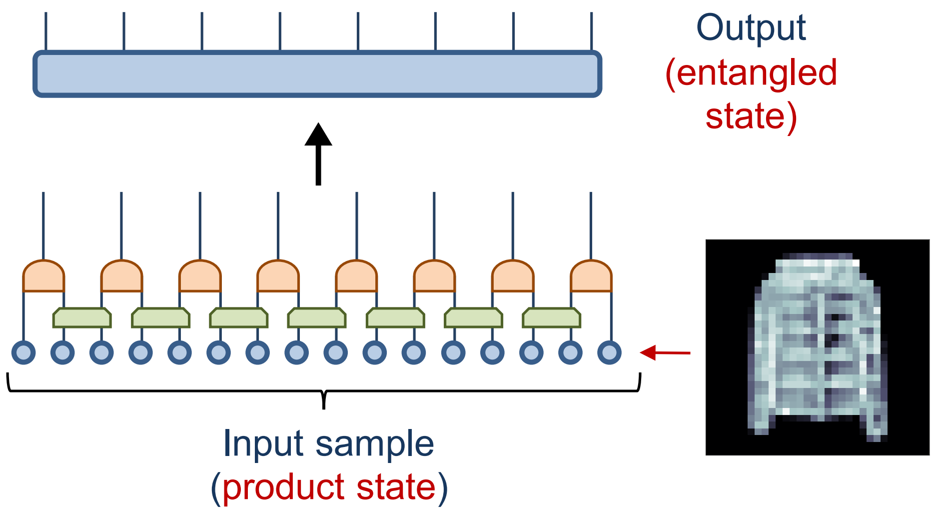
Tensor networks for machine learning

(2) Conceptual problem



TTN maps product states to product states

Each intermediate state can still be interpreted classically (i.e. as an image)



MERA maps product states to entangled states

Cannot relate intermediate states to classical data

Interpretability has been lost!

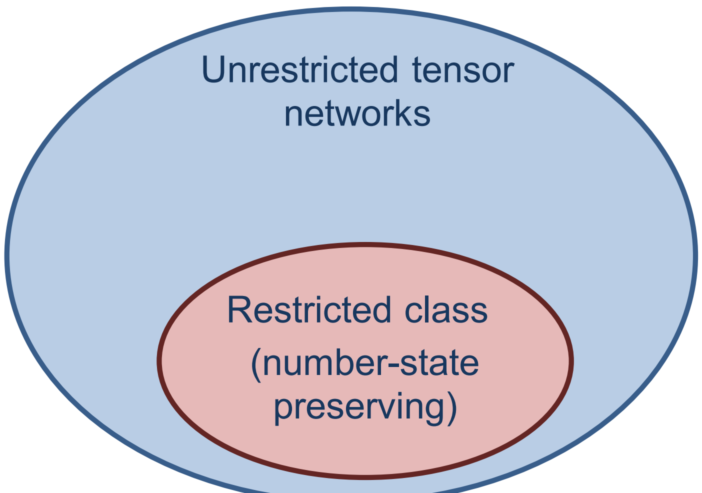
Restricted class of tensor networks

Number-State Preserving Tensor Networks as Classifiers for Supervised Learning
G.E., arXiv:1905.06352 (2019)

Can we fix these problems?

Can we identify a restricted class of tensor network suited for machine learning tasks?

should be efficiently contractible against product states	⇒	can be efficiently trained and applied as a classifier
should map product states into product states	⇒	preserves interpretation of states as classical data



Tensor networks are designed to represent complex superpositions of quantum states; they contain features that are not necessary if they are to be used as classifiers

$$|\Psi\rangle = c_0|\psi_0\rangle + c_1|\psi_1\rangle + c_2|\psi_2\rangle + \cdots$$

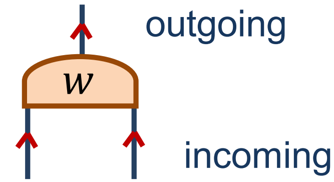
It makes sense that we should seek a restricted class of network that contains only the structure we need

My proposal:

Restrict to number state preserving tensors

Number state preserving tensors

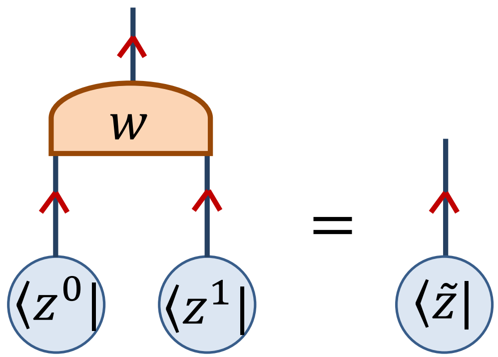
Consider tensor with oriented indices
(each index is incoming or outgoing):



Tensor is number-state preserving if it maps an input number-state to an output number-state

Number-state:
product state in the z-basis

Tensor notation

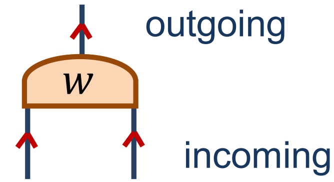


Mapping over all number states

$$\langle 0|\langle 0| \overset{w}{\mapsto} \langle 0|$$
$$\langle 1|\langle 0| \mapsto \langle 1|$$
$$\langle 0|\langle 1| \mapsto \langle 1|$$
$$\langle 1|\langle 1| \mapsto \langle 0|$$

Number state preserving tensors

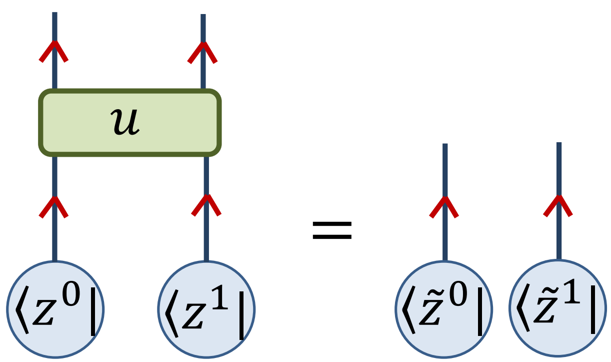
Consider tensor with oriented indices (each index is incoming or outgoing):



Tensor is number-state preserving if it maps an input number-state to an output number-state

Number-state: product state in the z-basis

Tensor notation



Mapping over all number states

$$u$$

$$\langle 0|\langle 0| \mapsto \langle 0|\langle 0|$$
$$\langle 1|\langle 0| \mapsto \langle 0|\langle 1|$$
$$\langle 0|\langle 1| \mapsto \langle 0|\langle 1|$$
$$\langle 1|\langle 1| \mapsto \langle 1|\langle 1|$$

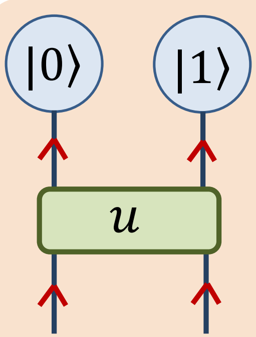
Reshaped as input-output matrix

output

input

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

- only non-zero element per row allowed
- can have multiple non-zero elements per column

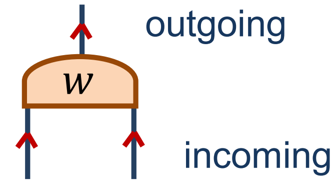


$$= \left(|0\rangle|1\rangle + |1\rangle|0\rangle\right)$$

Can generate entangled states when run in reverse!

Number state preserving tensors

Consider tensor with oriented indices (each index is incoming or outgoing):

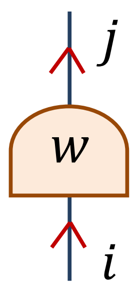


Tensor is number-state preserving if it maps an input number-state to an output number-state

Number-state: product state in the z-basis

number-state preserving $\Rightarrow$ Tensor has only one element per row when reshaped into an input-output matrix

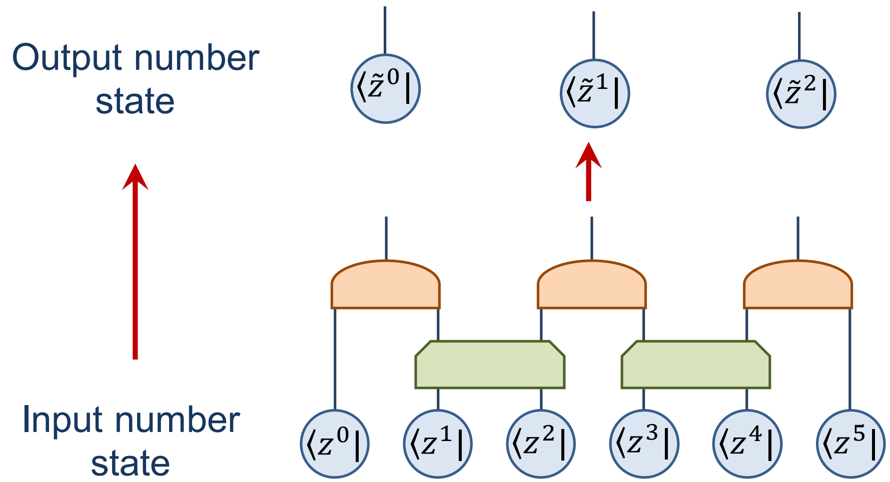
A final example:



$$w_{ij} = \begin{bmatrix} 1/3 & 0 & 0 \\ 0 & 0 & 3/4 \\ 0 & 0 & \sqrt{7}/4 \\ 2/3 & 0 & 0 \\ 2/3 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Tensor 'w' is both isometric and number state preserving

Number state preserving networks



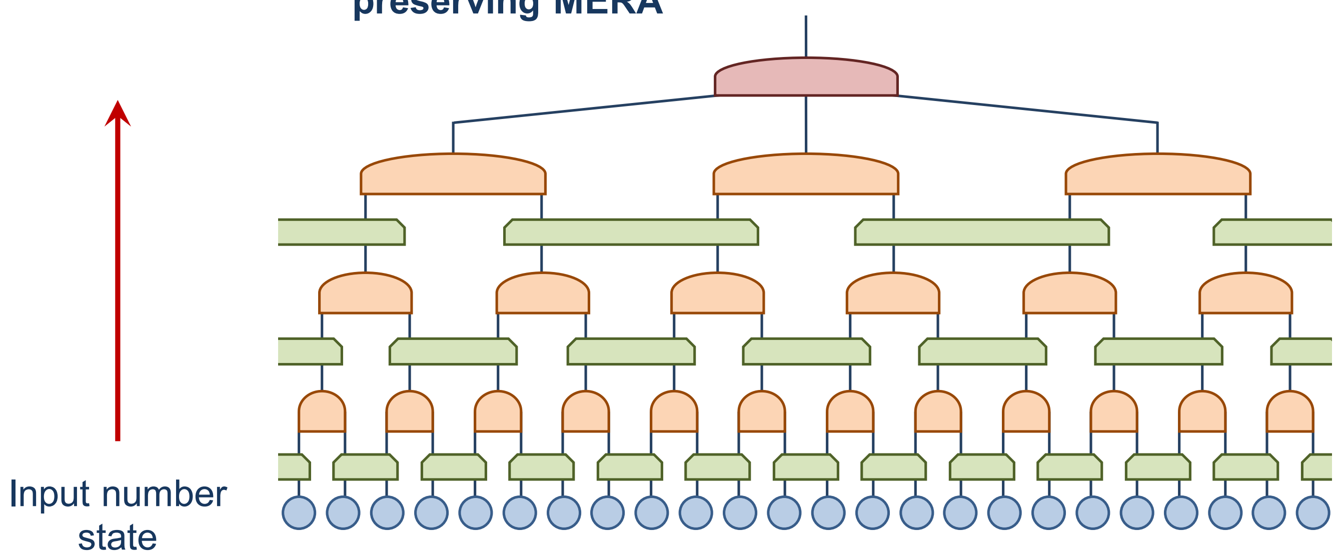
Products of number state preserving tensors are also number state preserving

(similar to how a product of isometric tensors is itself isometric)

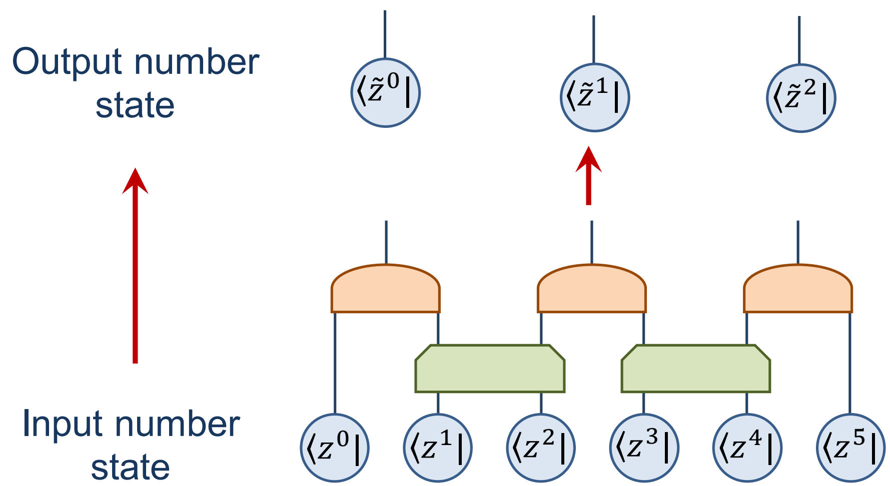
⇓

We can create number-state preserving versions of existing networks (e.g. MPS and MERA)

Number-state preserving MERA



Number state preserving networks

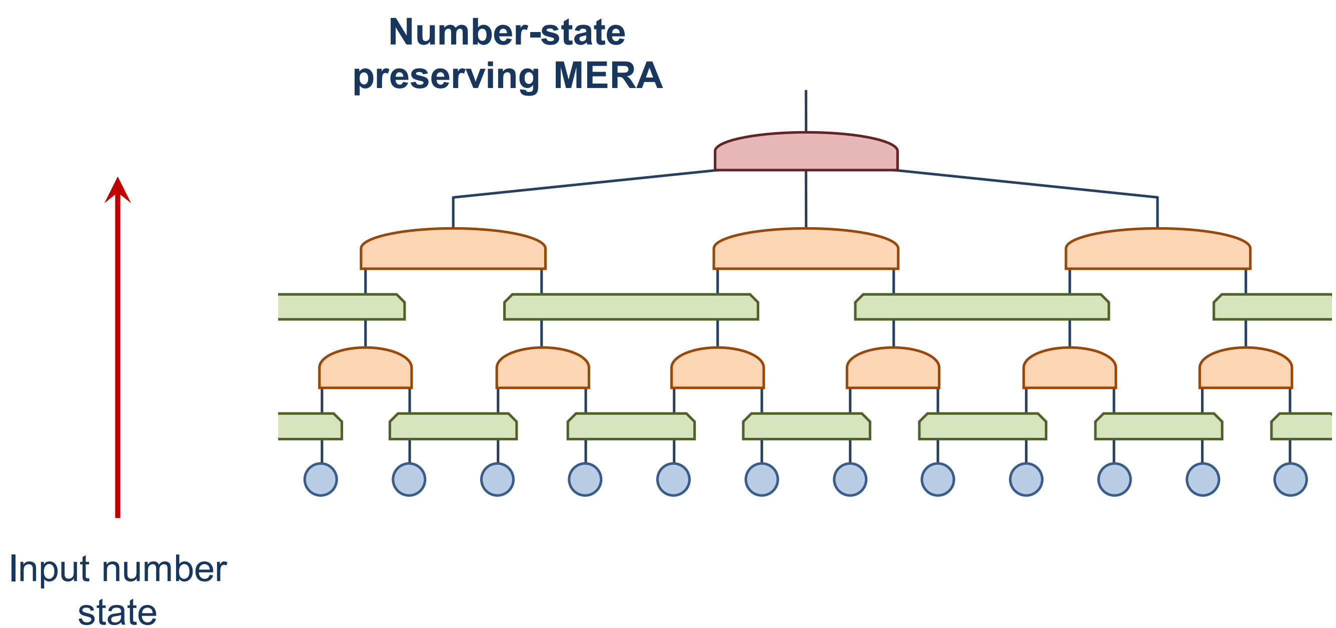


Products of number state preserving tensors are also number state preserving

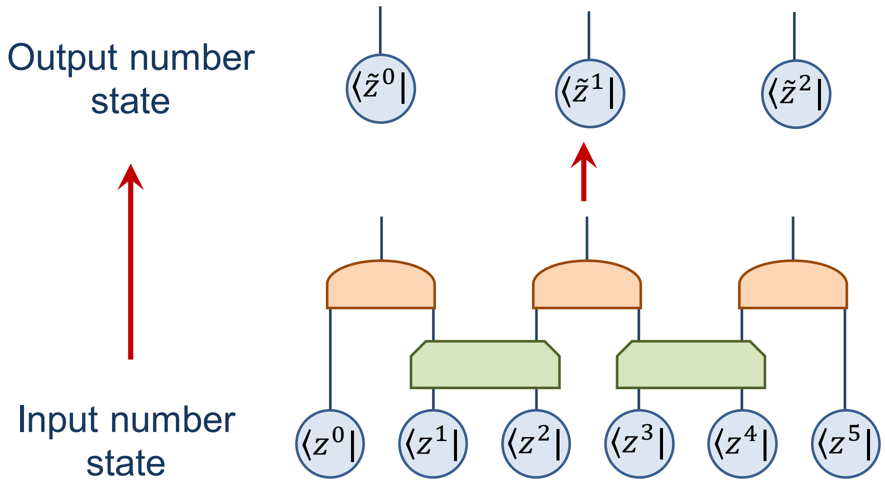
(similar to how a product of isometric tensors is itself isometric)

$\Downarrow$

We can create number-state preserving versions of existing networks (e.g. MPS and MERA)



Number state preserving networks



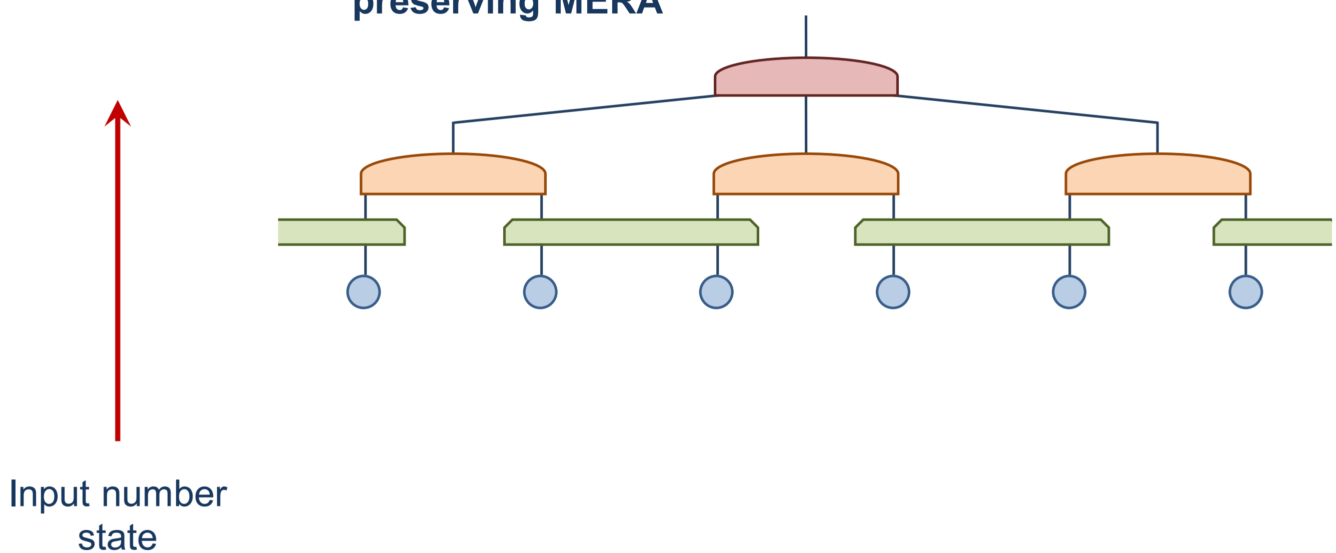
Products of number state preserving tensors are also number state preserving

(similar to how a product of isometric tensors is itself isometric)

⇓

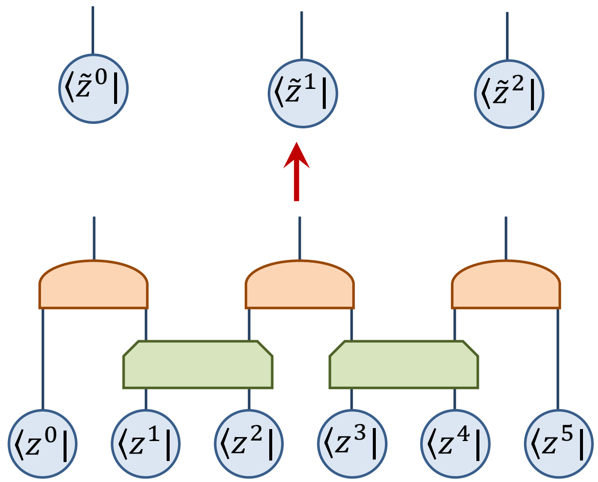
We can create number-state preserving versions of existing networks (e.g. MPS and MERA)

Number-state preserving MERA



Number state preserving networks

Output number state



Input number state

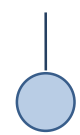
Products of number state preserving tensors are also number state preserving

(similar to how a product of isometric tensors is itself isometric)

$\Downarrow$

We can create number-state preserving versions of existing networks (e.g. MPS and MERA)

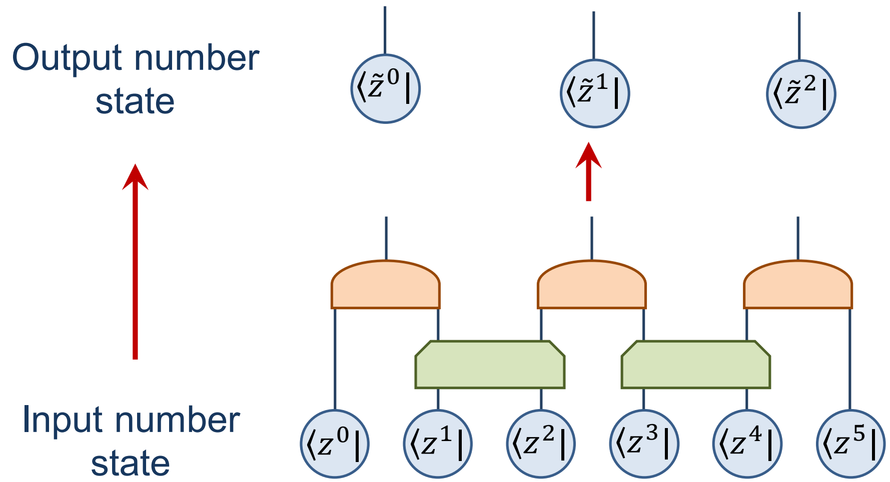
Number-state preserving MERA



Output number state

Input number state

Number state preserving networks



Products of number state preserving tensors are also number state preserving

(similar to how a product of isometric tensors is itself isometric)

$\Downarrow$

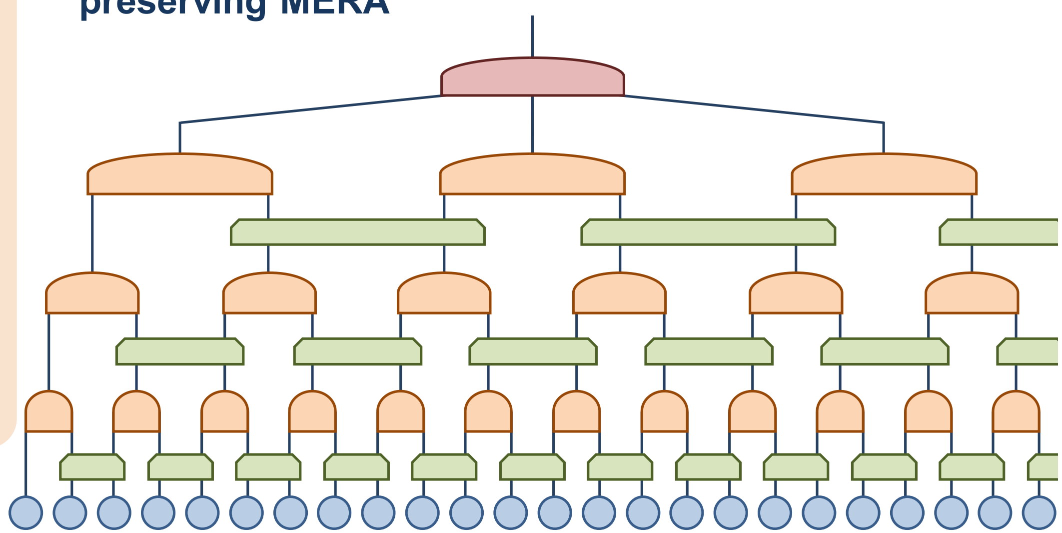
We can create number-state preserving versions of existing networks (e.g. MPS and MERA)

Useful restriction of tensor networks for machine learning?

Can efficiently be applied as a classifier (for classical data encoded as number states)

Preserves interpretability (intermediate states can still be understood as classical data)

Number-state preserving MERA



Number state preserving networks

Questions:

Is this restricted class of network powerful enough to interesting things? (i.e. can they describe entangled states?)

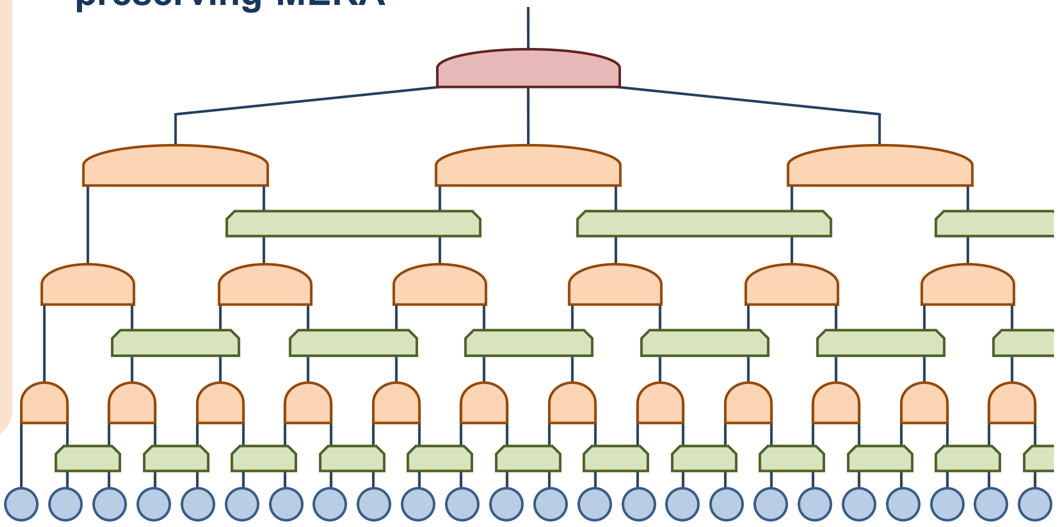
Can these networks be efficiently trained (or optimized) for machine learning tasks?

Useful restriction of tensor networks for machine learning?

Can efficiently be applied as a classifier (for classical data encoded as number states)

Preserves interpretability (intermediate states can still be understood as classical data)

Number-state preserving MERA



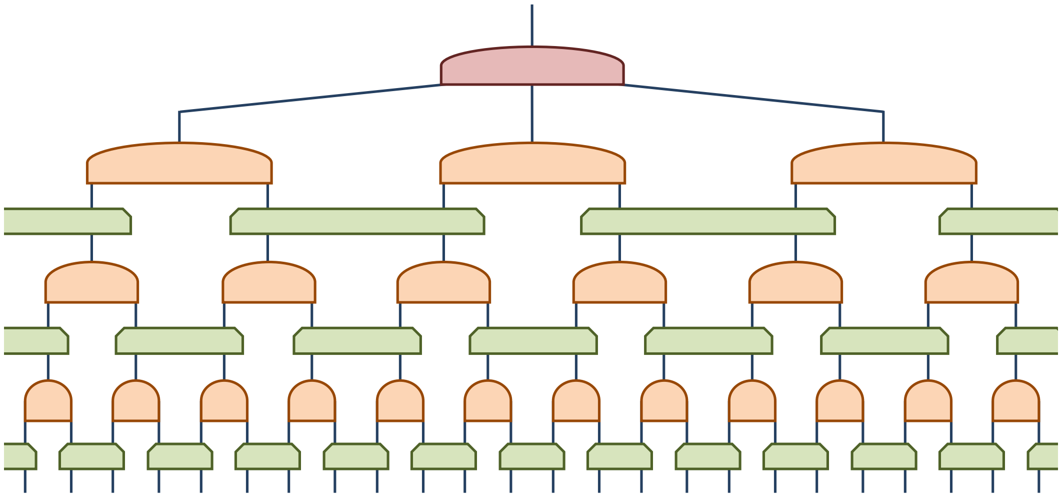
Number state preserving networks

Questions:

Is this restricted class of network powerful enough to interesting things? (i.e. can they describe entangled states?)

Can these networks be efficiently trained (or optimized) for machine learning tasks?

Number-state preserving MERA



product state

product state

product state

product state

product state

entangled state

entangled state

entangled state

Number state preserving networks

Number-state preserving MERA are non-trivial! If interpreted as describing a wavefunction ψ on the lattice:

- possess logarithmic scaling of entanglement entropy

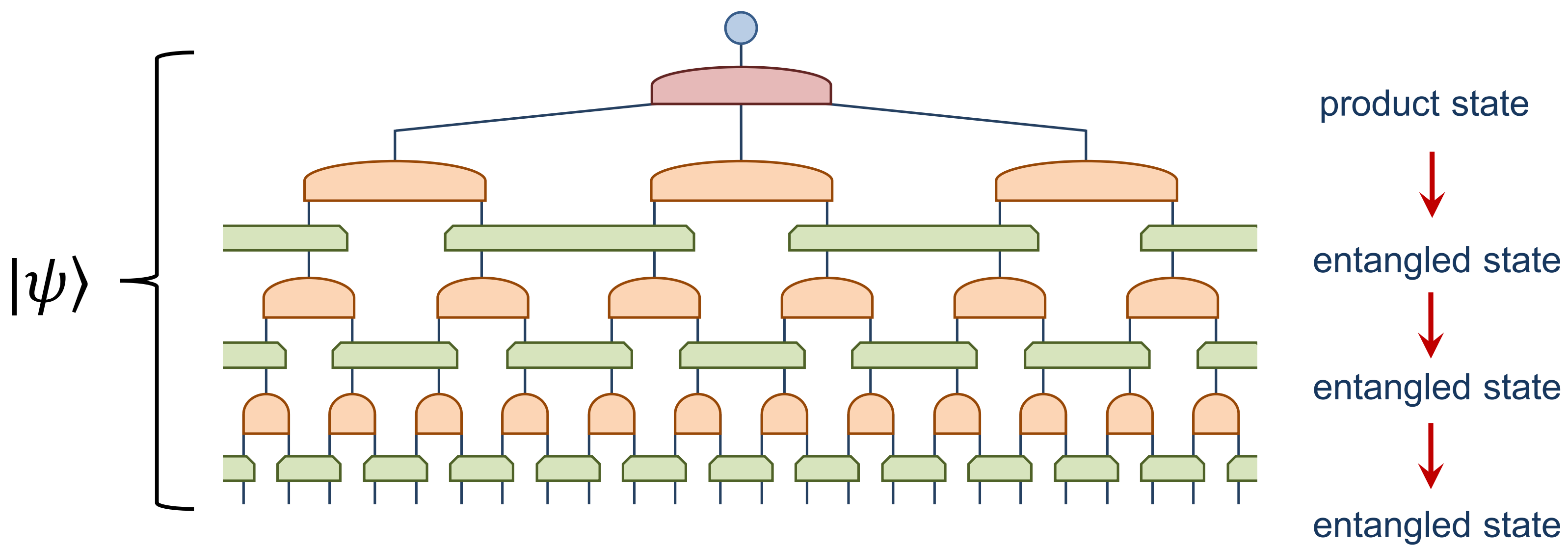
$$S_L = k_1 \log(L) + k_2$$

- possess polynomial correlation functions

Are they useful for describing quantum critical ground states?

Yes!
(sometimes…)

Number-state preserving MERA



Number state preserving networks

Number-state preserving MERA are non-trivial! If interpreted as describing a wavefunction ψ on the lattice:

- possess logarithmic scaling of entanglement entropy

$$S_L = k_1 \log(L) + k_2$$

- possess polynomial correlation functions

Are they useful for describing quantum critical ground states?

Yes!
(sometimes…)

Exact holographic tensor networks for the Motzkin spin chain

R.N. Alexander, G.E., I. Klich arXiv:1905.06352 (2019)

Motzkin and Fredkin models are described by a local interaction on 1D spin chain (open BC):

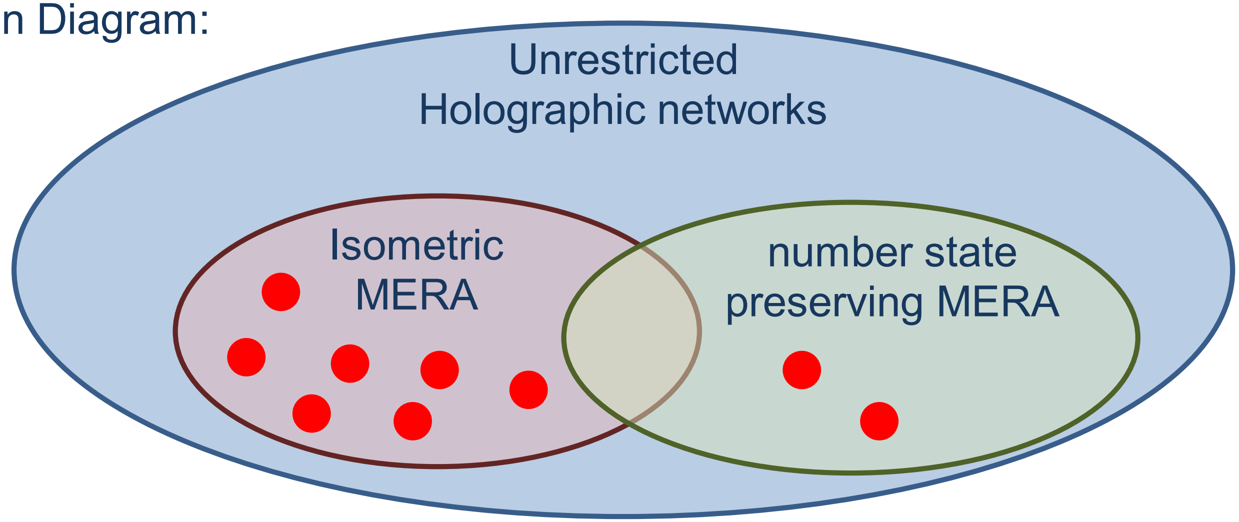
$$h_{[j,j+1,j+2]} = (1 + \sigma_j^z)(1 - \vec{\sigma}_{j+1} \cdot \vec{\sigma}_{j+2}) + (1 - \vec{\sigma}_j \cdot \vec{\sigma}_{j+1})(1 - \sigma_{j+2}^z)$$

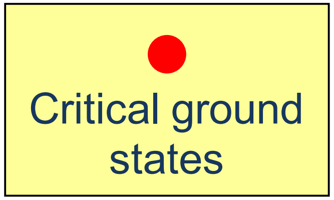
- possess unique ground state (but gapless excitations)
- ground states have logarithmic scaling of entanglement entropy

Exact description of ground states as number state preserving MERA!

Number state preserving networks

Venn Diagram:





Number state preserving MERA can describe interesting entangled states!

Open questions:

- What properties differ between isometric MERA and number state preserving MERA?
- What types of system can or cannot be described by number state preserving MERA?

Can these networks be efficiently trained (or optimized) for machine learning tasks?

Training tensor networks for classification tasks

Supervised learning task:

- assume we are given M training samples, each a vector of N integers, together with corresponding labels

samples

$$
\left.\begin{array}{l}
[0,0,1,0,1,0,1,1,0,0] \rightarrow 0 \\
[1,0,0,0,1,0,1,1,1,0] \rightarrow 1 \\
[0,0,0,0,0,0,1,1,0,0] \rightarrow 0 \\
[1,0,1,1,1,1,0,1,1,0] \rightarrow 1
\end{array}\right\}
$$

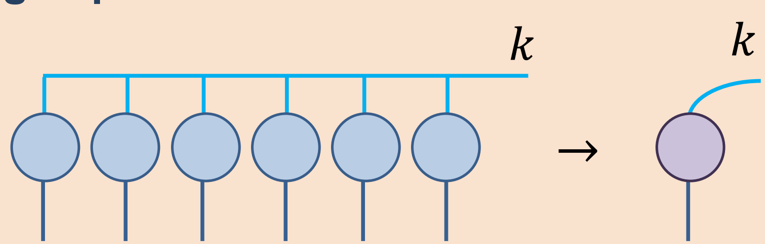
labels

$$\vdots$$

Goal: train a tensor network that matches samples to the correct labels

Tensor encoding of problem:

each sample is a number state (with k an index over samples)



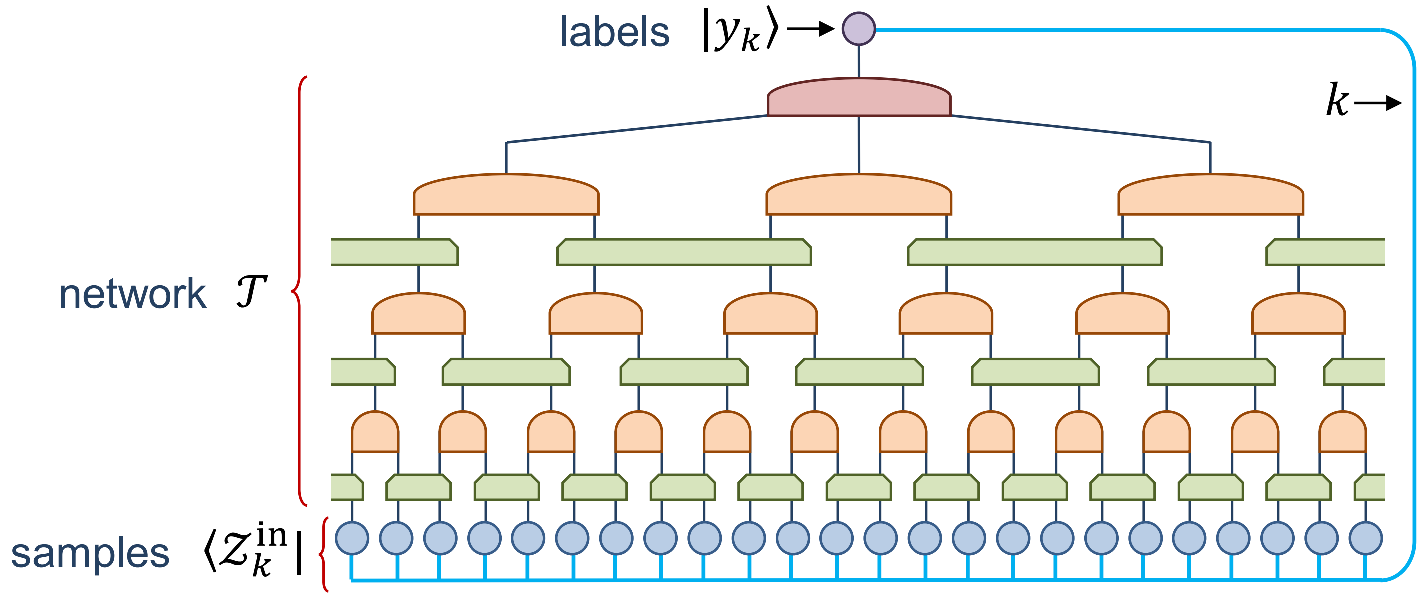
labels are single-site states

Training tensor networks for classification tasks

Can we quantify how well the tensor network matches training samples to their labels?

Use fidelity: $F = \sum_k \langle \mathcal{Z}_k^{\text{in}} | \mathcal{T} | y_k \rangle$

network matches samples to labels



Training the network for the supervised learning task = Optimizing the tensors in the network to maximize the fidelity

Use established methods (i.e. variational sweep):

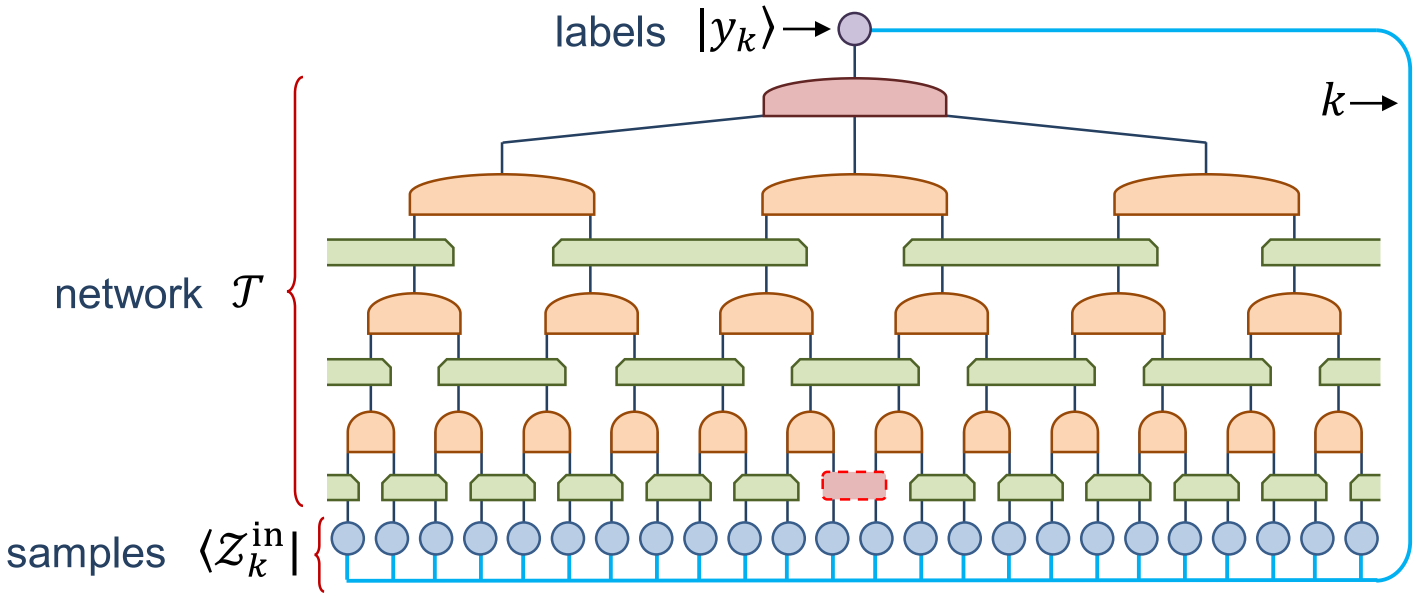
- update one tensor at a time (first computing the tensor environment)
- sweep over all tensors and iterate until converged

Training tensor networks for classification tasks

Can we quantify how well the tensor network matches training samples to their labels?

Use fidelity: $F = \sum_k \langle \mathcal{Z}_k^{\text{in}} | \mathcal{T} | y_k \rangle$

network matches samples to labels



Can we efficiently compute tensor environments?
(or the derivatives of the fidelity w.r.t each tensor

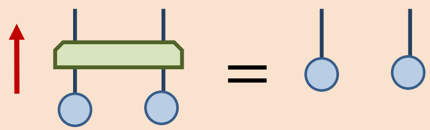
$$\frac{\partial}{\partial u} \langle Z_{\text{product}} | \psi_{\text{MERA}}(u, w) \rangle$$

Evaluation of environments

Can we efficiently compute derivatives?

$$\frac{\partial}{\partial u}\langle Z_{\text{product}}|\psi_{\text{MERA}}(u,w)\rangle$$

Simplifications due to number-state preserving tensors



⇓

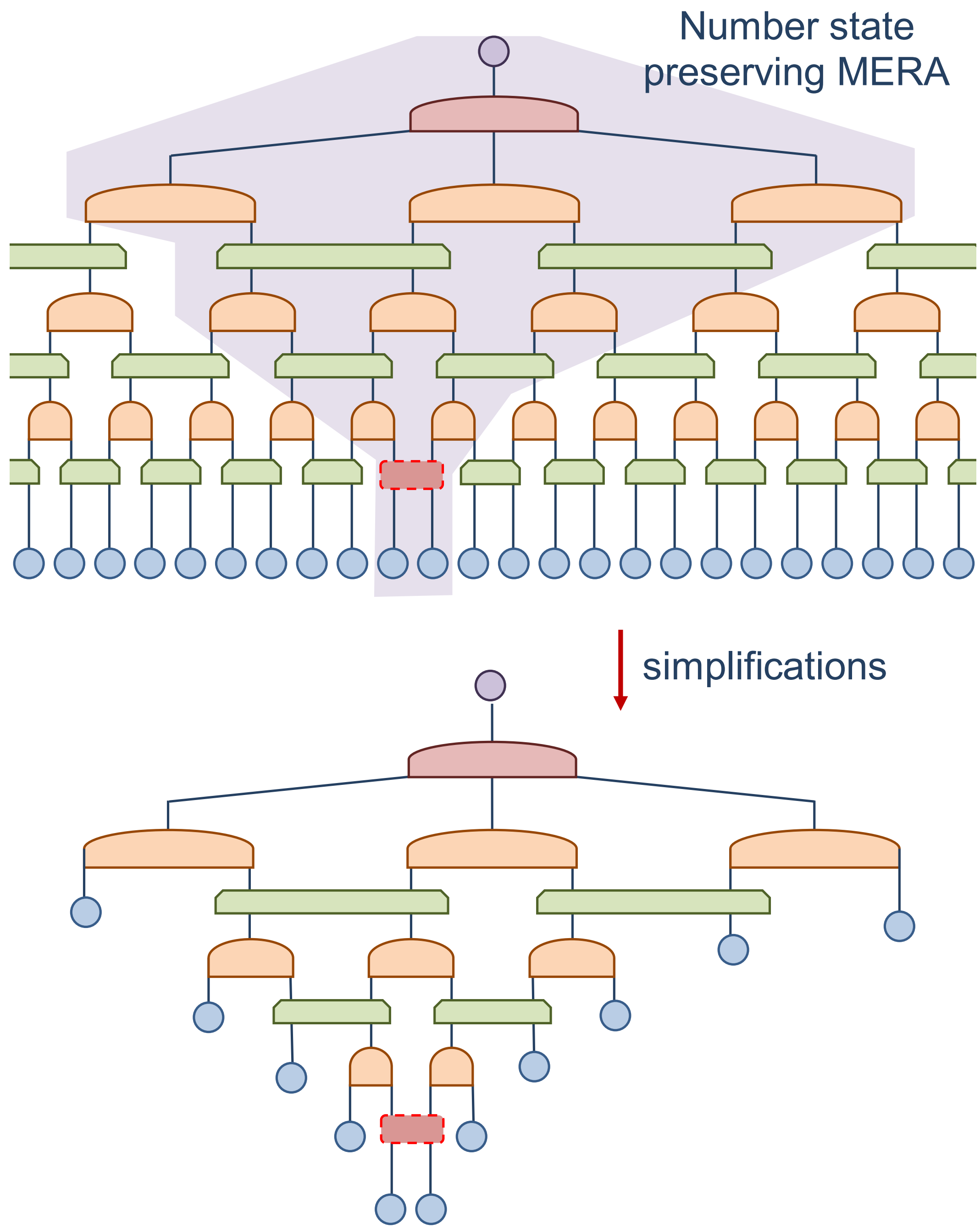
Simplified network has finite tree-width

⇓

Derivatives can be evaluated at cost: *O(log(N))*

⇓

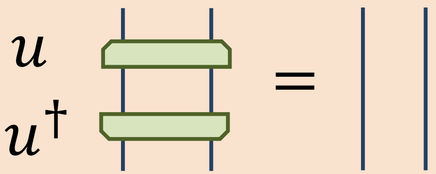
Network can be trained efficiently



Evaluation of environments

Isometric MERA:

Unitary / isometric constraints:



⇒

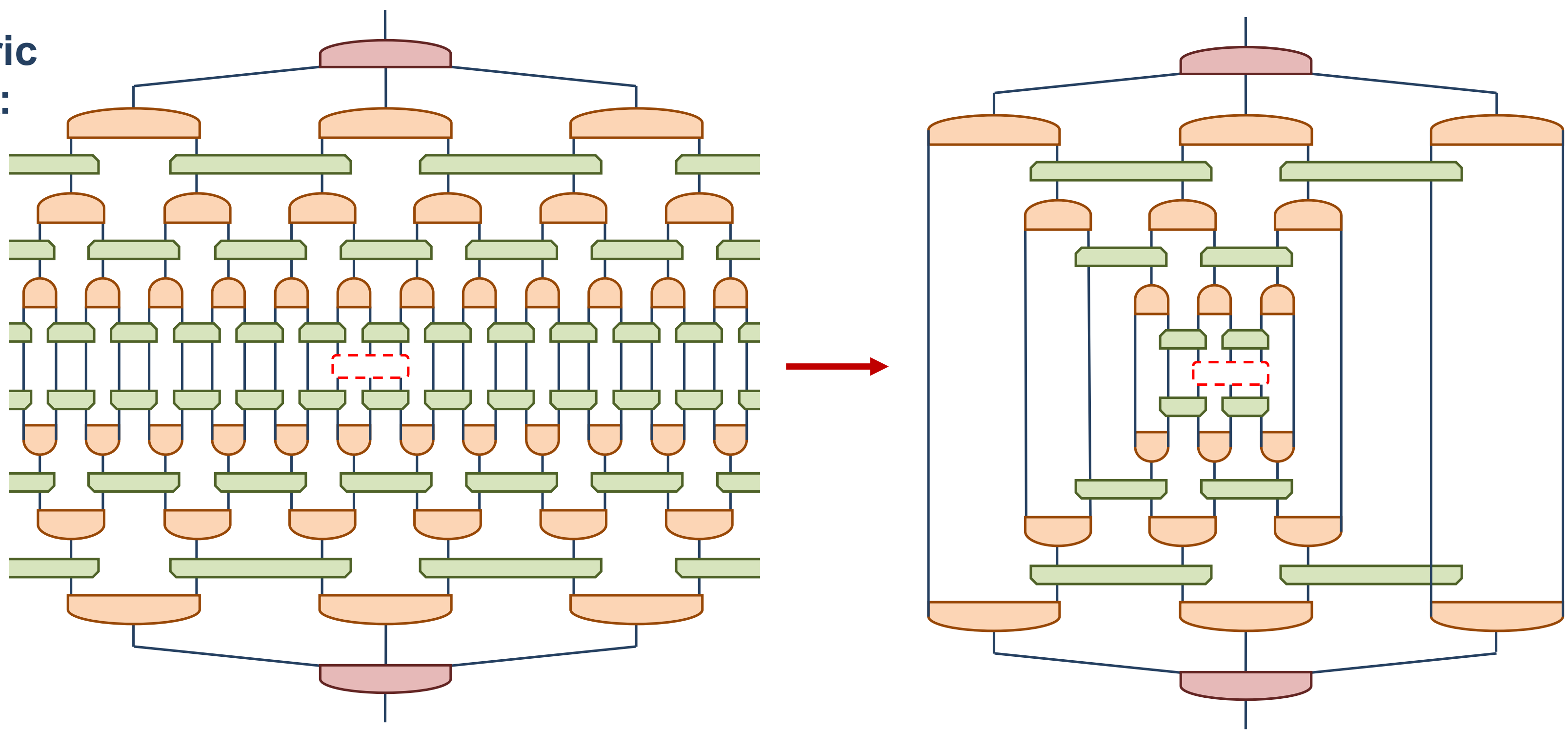
Simplifications occur when evaluating scalar product of MERA with itself:

$\langle \psi_{\text{MERA}} | h_{\text{local}} | \psi_{\text{MERA}} \rangle$

⇒

Efficient energy minimization to find the ground state of local Hamiltonians

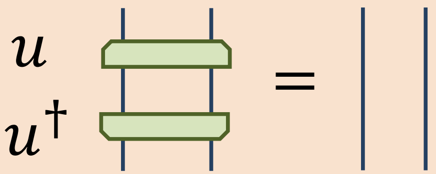
Isometric MERA:



Evaluation of environments

Isometric MERA:

Unitary / isometric constraints:



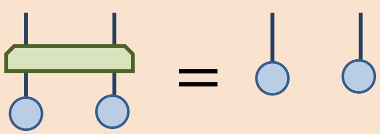
$\Rightarrow$ Simplifications occur when evaluating scalar product of MERA with itself:

$\langle \psi_{\mathrm{MERA}} | h_{\mathrm{local}} | \psi_{\mathrm{MERA}} \rangle$

$\Rightarrow$ Efficient energy minimization to find the ground state of local Hamiltonians

Number state preserving MERA:

Number-state preserving constraints:



$\Rightarrow$ Simplifications occur when evaluating scalar product of MERA with a product state

$\langle Z_{\mathrm{product}} | h_{\mathrm{local}} | \psi_{\mathrm{MERA}} \rangle$

$\Rightarrow$ Efficient training to maximize fidelity against ensemble of classical data

Number state preserving MERA seem to be a natural choice for machine learning tasks!

Benchmark problem

So far:
- We have proposed a restricted class of tensor network state
- Argued that this class can still possess interesting entanglement
- Argues that this class can be efficiently applied as classifiers

How well do these ideas work in practice?

We should crawl before we try to walk. Lets consider a toy supervised learning problem

Benchmark problem

Number-State Preserving Tensor Networks as Classifiers for Supervised Learning
G.E., arXiv:1905.06352 (2019)

Height classification problem:

- Each 'pixel' of a sample is in state: $z \in \{+, 0, -\}$
- Samples are labelled by whether the sum (under regular addition) of all pixels is positive, zero, or negative

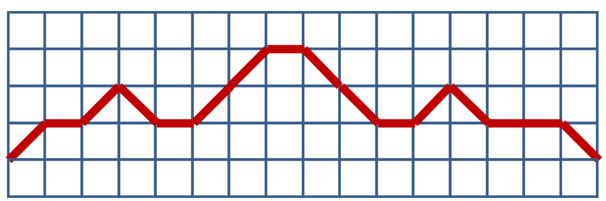
samples

$$\left.\begin{array}{l} [+, 0, -, 0, -, 0, +, +, 0, 0] \rightarrow \quad + \\ [0, +, +, 0, 0, 0, -, -, -, 0] \rightarrow \quad - \\ [-, +, 0\ +, +, -, -, 0, -, +] \rightarrow \quad 0 \end{array}\right\}$$

labels

$$\vdots$$

Related to height models (think of each sample as a path)



Why this problem? Given an ensemble of samples, the block entropy scales logarithmically:

$\Rightarrow$ Tree Tensor Network should not work so well

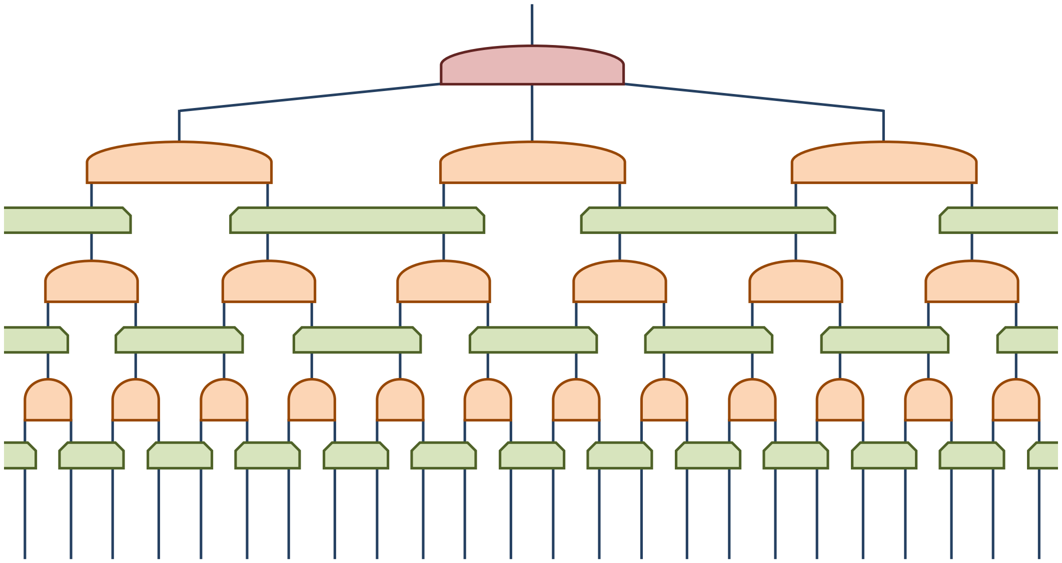
$\Rightarrow$ MERA could work well

Benchmark problem

Number-State Preserving Tensor Networks as Classifiers for Supervised Learning

G.E., arXiv:1905.06352 (2019)

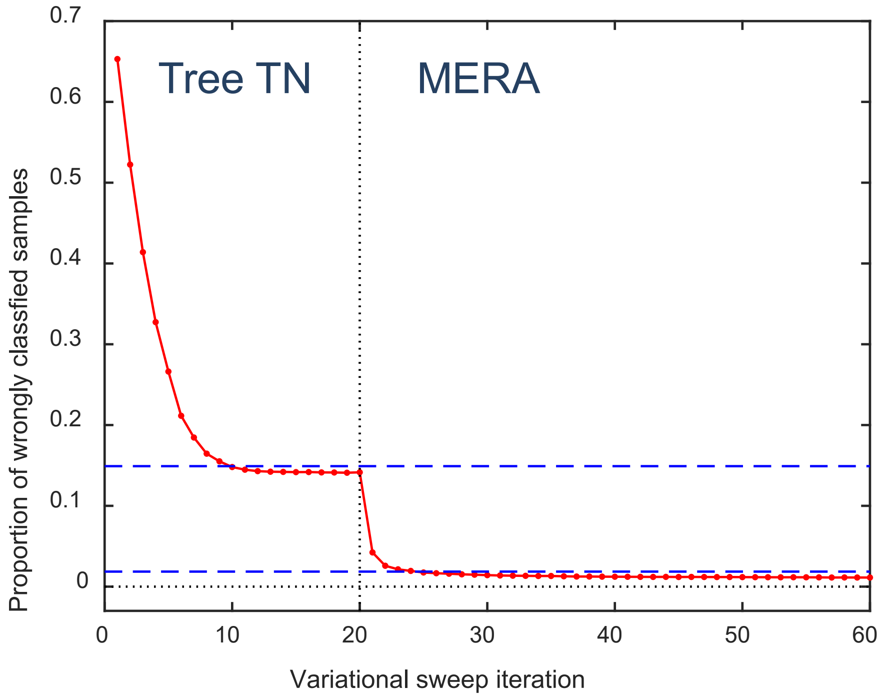
- Chain of N = 24 sites (⇒ 3^{24} basis states)
- Generate 12000 random training samples
- Train 3-level MERA as a classifier (bond dimension chi = 9)
- Initialize disentanglers as identity, initialize other tensors randomly
- After training, generate new samples to test the accuracy as a classifier



Benchmark problem

Number-State Preserving Tensor Networks as Classifiers for Supervised Learning
G.E., arXiv:1905.06352 (2019)

- Chain of N = 24 sites (⇒ 3^{24} basis states)
- Generate 12000 random training samples
- Train 3-level MERA as a classifier (bond dimension chi = 9)
- Initialize disentanglers as identity, initialize other tensors randomly
- After training, generate new samples to test the accuracy as a classifier



Training results (averaged over many runs)

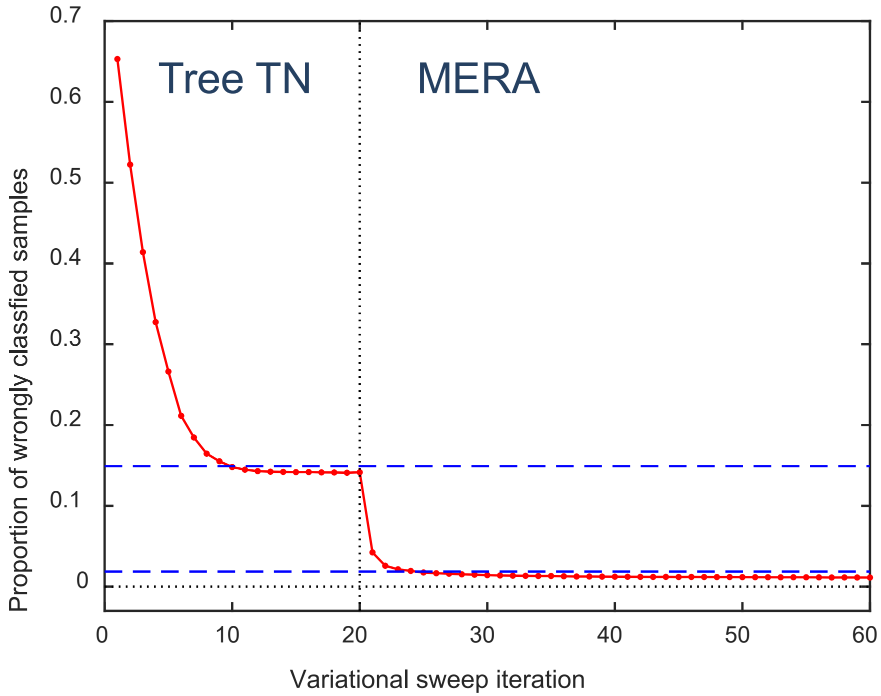
Wrongly classified training samples: Tree TN: 14%, MERA: 1%

Wrongly classified test samples: Tree TN: 15%, MERA: 2%

Benchmark problem

Number-State Preserving Tensor Networks as Classifiers for Supervised Learning
G.E., arXiv:1905.06352 (2019)

- Optimization algorithm generally converges well and can easily be scaled to larger system sizes and higher bond dimensions
- Disentanglers have a significant effect (MERA is vastly more accurate than a tree TN)
- Good generalization from training to test samples (i.e. we are not just overfitting to the training data)

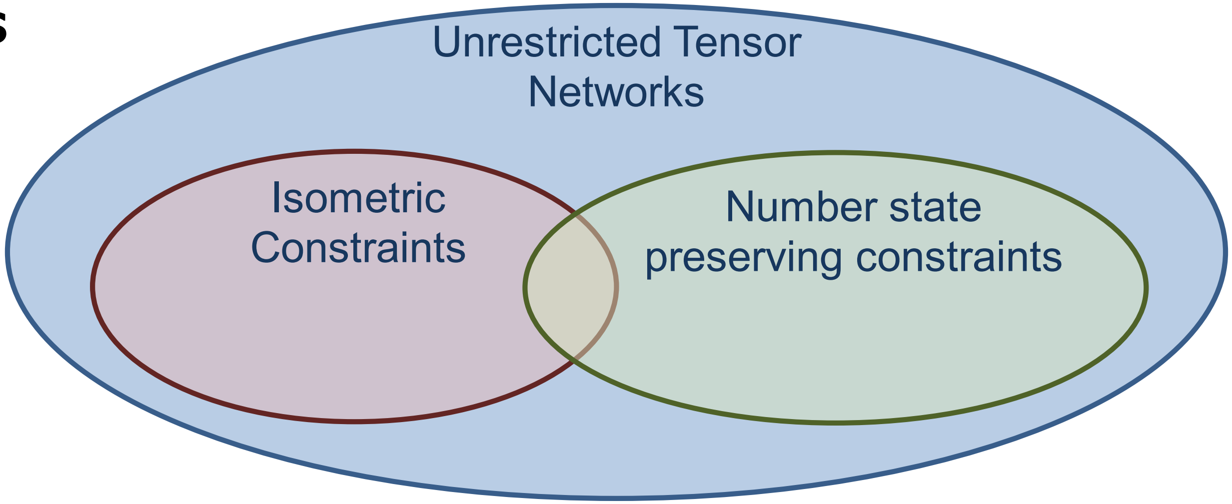


Training results (averaged over many runs)

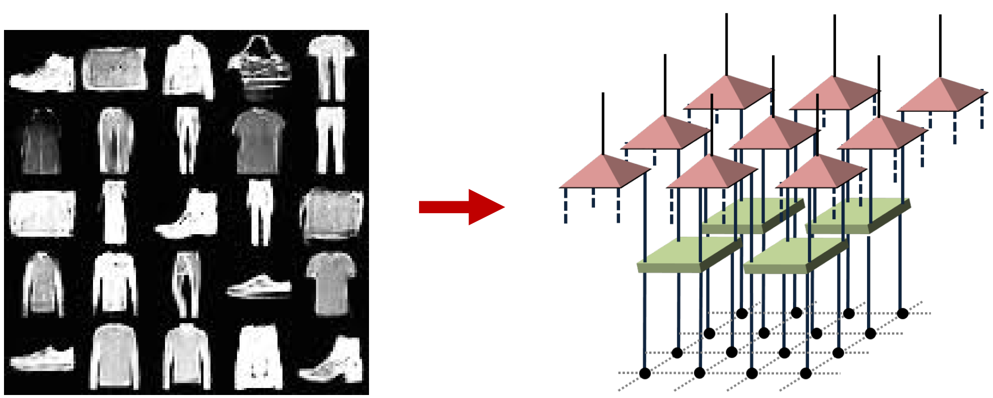
Wrongly classified training samples: Tree TN: 14%, MERA: 1%

Wrongly classified test samples: Tree TN: 15%, MERA: 2%

Conclusions



- number state preserving tensors seems to be a natural restriction for tensor networks when applying to classification problems
- we have efficient training algorithms for this class of network
- performs well in toy classification problems



Can we apply 2D MERA to difficult problems? In progress…

Thanks!