Title: Classical and Quantum Machine Learning with Tensor Networks

Date: May 18, 2018 03:30 PM

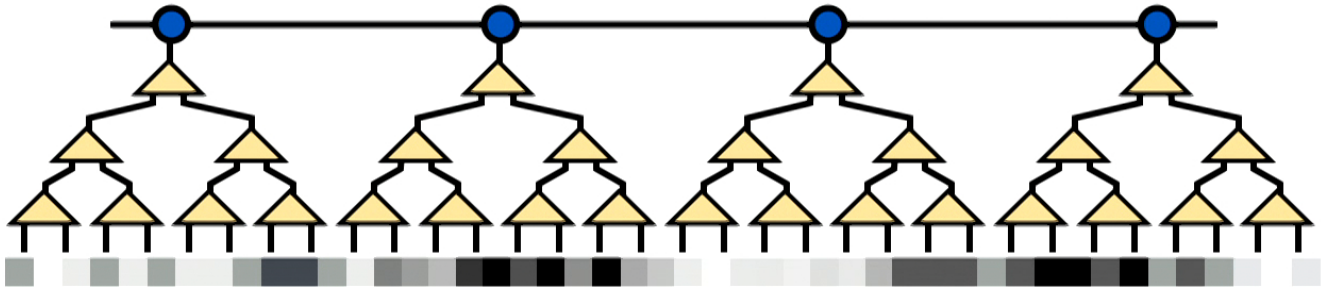
URL: http://pirsa.org/18050061

Abstract: Over the last decade, there have been enormous gains in machine learning technology primarily driven by neural networks. A major reason neural networks have outperformed older techniques is that the cost of optimizing them scales well with the size of the training dataset. But neural networks have the drawback that they are not very well understood theoretically.

Recent work by several groups has explored an alternative approach to creating machine learning model functions based on tensor networks, which are a technique for parameterizing many-body quantum wavefunctions. The cost of training tensor network models scales similarly to the cost of training neural networks. In addition, their relatively simple, linear structure has provided good theoretical understanding of their properties, and underpins many powerful techniques to optimize and manipulate them.

After introducing tensor network machine learning models, I will discuss some of the techniques to optimize them and results for supervised and generative machine learning tasks. These techniques can automatically adapt the number of parameters and suggest interesting interpretations and extensions for 'deep' tensor network variants. I will conclude by discussing a recent tensor network based proposal to formulate hybrid quantum-classical algorithms for machine learning with quantum computers.

Classical and Quantum Machine Learning with Tensor Networks



E.M. Stoudenmire

May 2018 - Perimeter

FLATIRON INSTITUTE
SIMONS FOUNDATION

Flatiron Institute

FLATIRON INSTITUTE

SIMONS FOUNDATION

The mission of the Flatiron Institute is to advance scientific research through computational methods, including data analysis, modeling and simulation.

CCA: Center for Computational Astrophysics

CCB: Center for Computational Biology

CCQ: Center for Computational Quantum Physics

Plus fourth center to be decided

Overview

Tensor networks = powerful model of
many-body wavefunctions

Overview

Tensor networks = powerful model of
many-body wavefunctions

Tensor networks can parameterize
machine learning models too

Overview

Tensor networks = powerful model of
many-body wavefunctions

Tensor networks can parameterize
machine learning models too

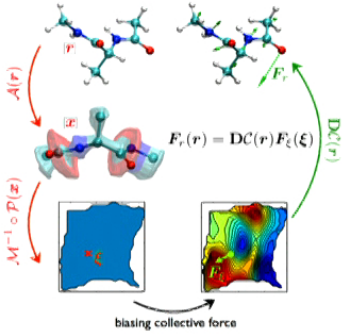
Resulting learning architecture has interesting
capabilities, connections to quantum computing

Machine learning galvanizing industry & science

Language Processing

Self-driving cars

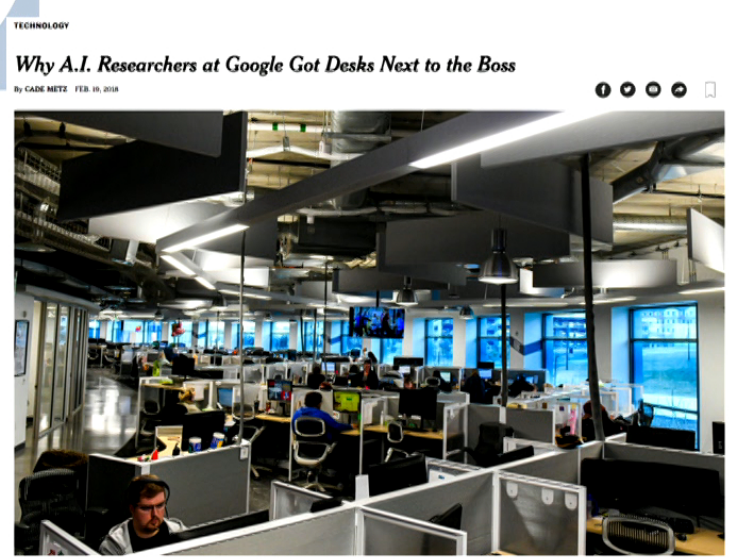
Medicine



Materials Science / Chemistry

Google rebranded a "machine learning first company"

Neural nets replace linguistic approach to Google Translate



arXiv.org > quant-ph > arXiv:1802.06002

Quantum Physics

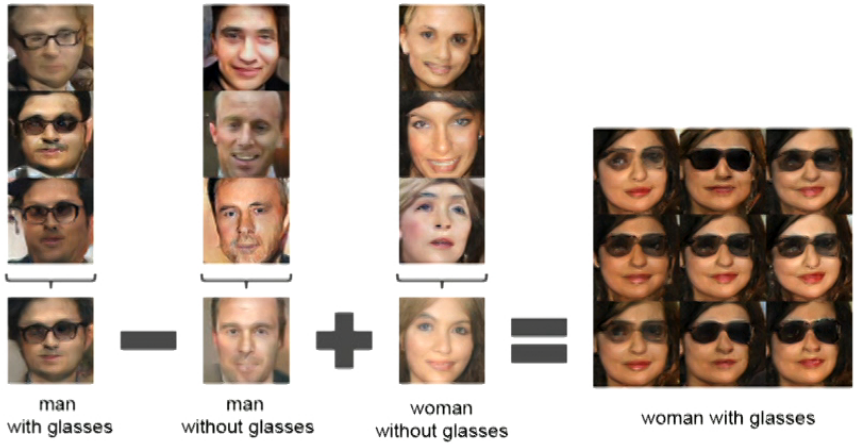
Classification with Quantum Neural Networks on Near Term Processors

Edward Farhi, Hartmut Neven

(Submitted on 16 Feb 2018)

Quantum machine learning

Doing tasks thought impossible...

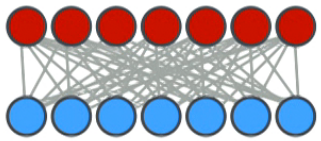


UNSUPERVISED REPRESENTATION LEARNING
WITH DEEP CONVOLUTIONAL
GENERATIVE ADVERSARIAL NETWORKS

Alec Radford & Luke Metz
indico Research
Boston, MA
{alec,luke}@indico.io

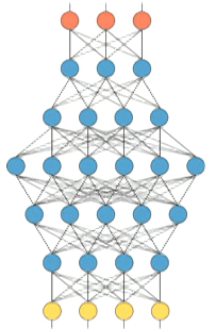
Soumith Chintala
Facebook AI Research
New York, NY
soumith@fb.com

Machine learning has physics in its DNA

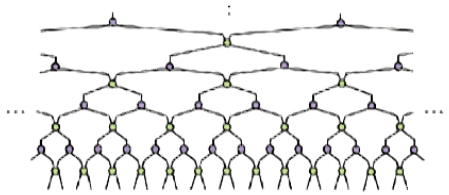


Boltzmann Machines

Disordered Ising Model



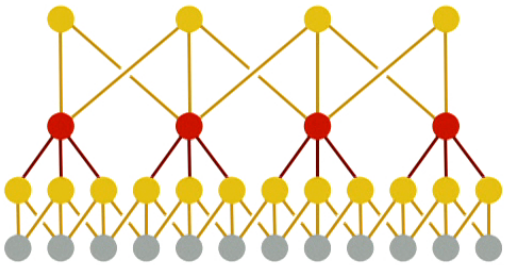
Deep Belief Networks



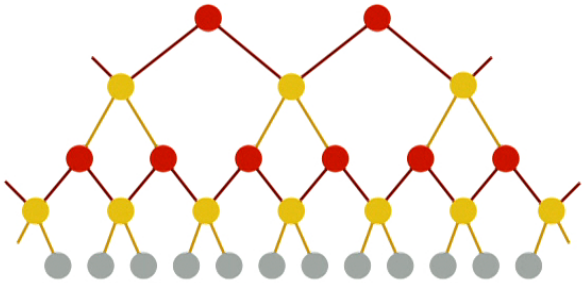
The "Renormalization Group"

P. Mehta and D.J. Schwab, arxiv:1410.3831
S. Bradde and W. Bialek, arxiv:1610.09733

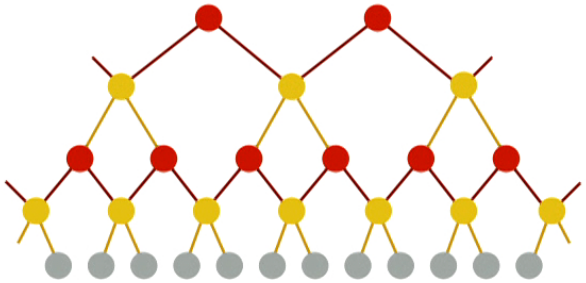
Convolutional neural network



"MERA" tensor network



Are tensor networks useful for machine learning?

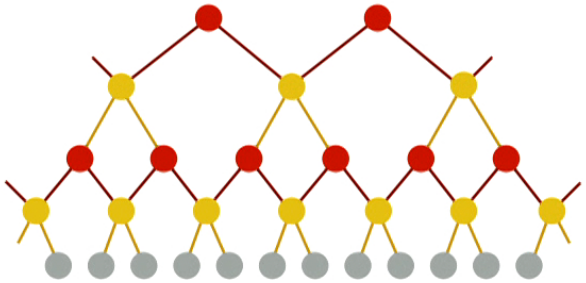


"MERA" tensor network

Tensor networks can represent weights of useful and interesting machine learning models

Realized benefits:

Are tensor networks useful for machine learning?



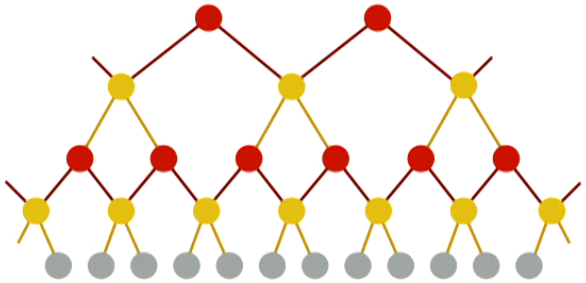
"MERA" tensor network

Tensor networks can represent weights of useful and interesting machine learning models

Realized benefits:

- Linear scaling

Are tensor networks useful for machine learning?



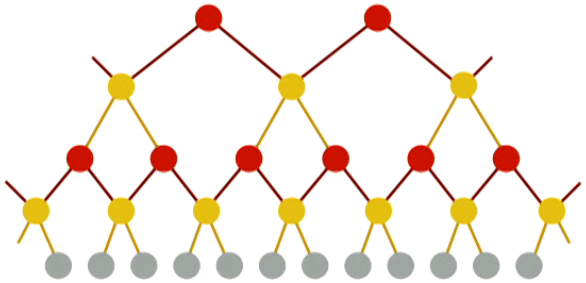
"MERA" tensor network

Tensor networks can represent weights of useful and interesting machine learning models

Realized benefits:

- Linear scaling
- Adaptive weights

Are tensor networks useful for machine learning?



"MERA" tensor network

Tensor networks can represent weights of useful and interesting machine learning models

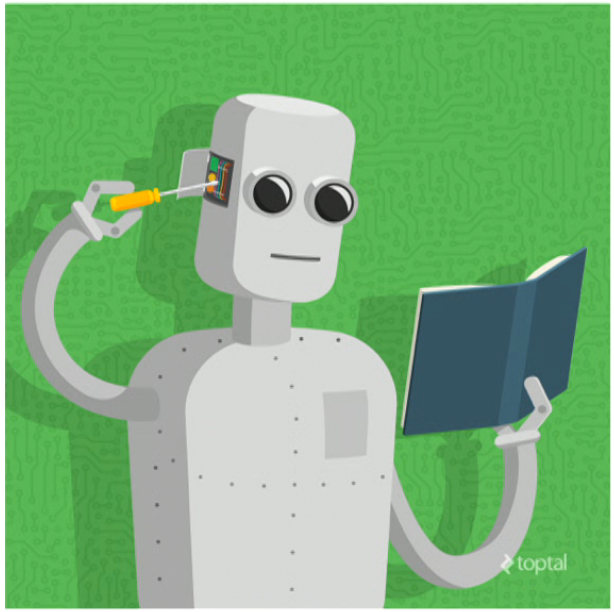
Realized benefits:

- Linear scaling
- Adaptive weights
- Learning data "features"

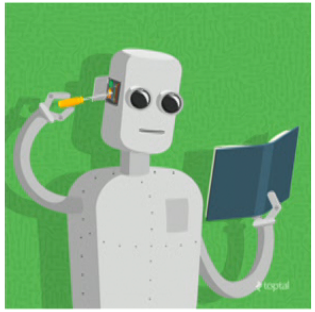
Future benefits?

- Interpretability / theory

What is machine learning?

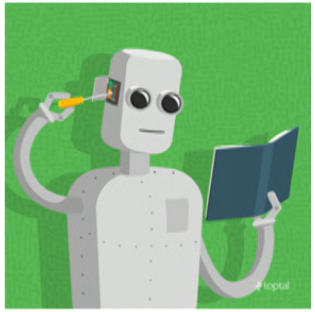


What is machine learning?



Data driven problem solving

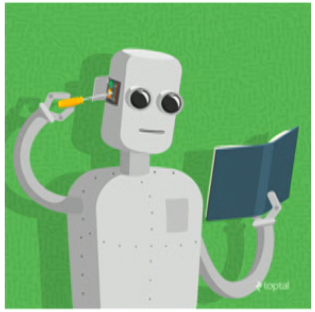
What is machine learning?



Data driven problem solving

Any system that, given more data, performs increasingly better at some task

What is machine learning?



Data driven problem solving

Any system that, given more data, performs increasingly better at some task

Framework / philosophy, not single method

Goal: train model $f(\mathbf{x})$

mapping input $\mathbf{x}$ to target

$$f(\mathbf{x}) \longrightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{"dog"}$$

Goal: train model $f(\mathbf{x})$

mapping input $\mathbf{x}$ to target

$$f(\mathbf{x}) \longrightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{"dog"}$$

Goal: train model $f(\mathbf{x})$

mapping input $\mathbf{x}$ to target

$$f(\mathbf{x}) \longrightarrow \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{ "cat"}$$

Philosophy of Machine Learning

- Map from images to labels is just a function

Philosophy of Machine Learning

- Map from images to labels is just a function

- Parameterize a set of very flexible functions
  (prefer convenient functions over "correct" ones)

Philosophy of Machine Learning

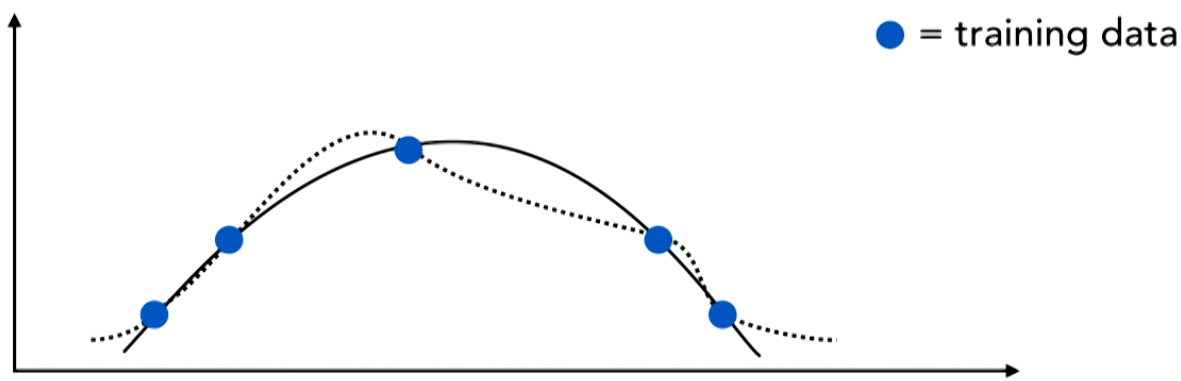
- Map from images to labels is just a function

- Parameterize a set of very flexible functions
  (prefer convenient functions over "correct" ones)

- Prevent overfitting by regularization (prefer simple functions)

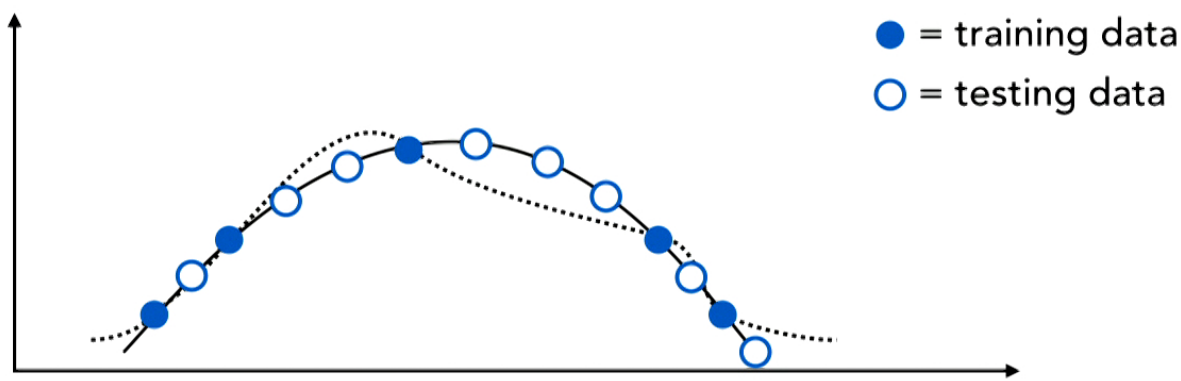
Philosophy of Machine Learning

- Map from images to labels is just a function
- Parameterize a set of very flexible functions
  (prefer convenient functions over "correct" ones)
- Prevent overfitting by regularization (prefer simple functions)



Philosophy of Machine Learning

- Map from images to labels is just a function
- Parameterize a set of very flexible functions
  (prefer convenient functions over "correct" ones)
- Prevent overfitting by regularization (prefer simple functions)



Supervised Learning

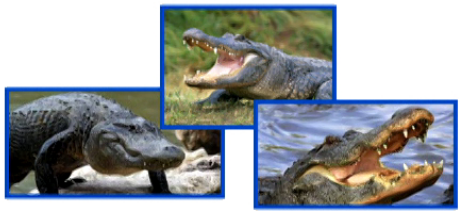
Given labeled training data (labels A and B)

Find *decision function* $f(\mathbf{x})$

$$f(\mathbf{x}) > 0 \qquad \mathbf{x} \in A$$

$$f(\mathbf{x}) < 0 \qquad \mathbf{x} \in B$$

Example: identify photos of alligators and bears



Unsupervised Learning

Given unlabeled training data $\{\mathbf{x}_j\}$

Unsupervised Learning

Given unlabeled training data $\{\mathbf{x}_j\}$

- Find function $f(\mathbf{x})$ such that $f(\mathbf{x}_j) \simeq p(\mathbf{x}_j)$
- Find function $f(\mathbf{x})$ such that $|f(\mathbf{x}_j)|^2 \simeq p(\mathbf{x}_j)$

Unsupervised Learning

Given unlabeled training data $\{\mathbf{x}_j\}$

- Find function $f(\mathbf{x})$ such that $f(\mathbf{x}_j) \simeq p(\mathbf{x}_j)$
- Find function $f(\mathbf{x})$ such that $|f(\mathbf{x}_j)|^2 \simeq p(\mathbf{x}_j)$
- Find data clusters and which data belongs to each cluster

Unsupervised Learning

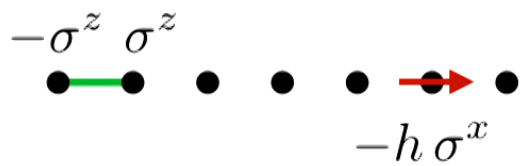
Given unlabeled training data $\{\mathbf{x}_j\}$

- Find function $f(\mathbf{x})$ such that $f(\mathbf{x}_j) \simeq p(\mathbf{x}_j)$
- Find function $f(\mathbf{x})$ such that $|f(\mathbf{x}_j)|^2 \simeq p(\mathbf{x}_j)$
- Find data clusters and which data belongs to each cluster
- Discover reduced representations of data for other learning tasks (e.g. supervised)

What are Tensor Networks?

Original setting is quantum mechanics

Spin model (transverse field Ising model):



$$\hat{H} = \sum_j (-\sigma_j^z \sigma_{j+1}^z - h\,\sigma_j^x)$$

$\uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow$ $\qquad h \ll 1$

$\rightarrow \quad \rightarrow \quad \rightarrow \quad \rightarrow \quad \rightarrow \quad \rightarrow \quad \rightarrow$ $\qquad h \gg 1$

Wavefunction just a rule to
map spin configurations to numbers $\Psi^{s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_8}$

$$\uparrow\downarrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow \longrightarrow \Psi^{\uparrow\downarrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow}$$

$$\uparrow\downarrow\uparrow\downarrow\uparrow\uparrow\uparrow\uparrow \longrightarrow \Psi^{\uparrow\downarrow\uparrow\downarrow\uparrow\uparrow\uparrow\uparrow}$$

$$\uparrow\uparrow\uparrow\uparrow\downarrow\uparrow\downarrow\uparrow \longrightarrow \Psi^{\uparrow\uparrow\uparrow\uparrow\downarrow\uparrow\downarrow\uparrow}$$

$$\uparrow\downarrow\downarrow\downarrow\downarrow\uparrow\uparrow\uparrow \longrightarrow \Psi^{\uparrow\downarrow\downarrow\downarrow\downarrow\uparrow\uparrow\uparrow}$$

Simplest rule: store every amplitude separately

Let's make a different rule

Introduce matrices, one for each spin

$$\uparrow \longrightarrow M^{\uparrow} = \begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \end{bmatrix}$$

$$\downarrow \longrightarrow M^{\downarrow} = \begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \end{bmatrix}$$

Östlund, Rommer, Phys. Rev. Lett. **75**, 3537 (1995)

Compute wavefunction by multiplying matrices together

$$\Psi^{\uparrow\downarrow\uparrow\uparrow\downarrow} \approx$$

Compute wavefunction by multiplying matrices together

$$\Psi^{\uparrow\downarrow\uparrow\uparrow\downarrow} \approx M_1^{\uparrow} M_2^{\downarrow} M_3^{\uparrow} M_4^{\uparrow} M_5^{\downarrow}$$

This rule is called a *matrix product state* (MPS)

$$\Psi^{s_1 s_2 s_3 s_4 s_5} = M_1^{s_1} M_2^{s_2} M_3^{s_3} M_4^{s_4} M_5^{s_5}$$

- Size of matrices called m (the "bond dimension")
- For m = $2^{N/2}$ can represent any state of N spins
- Just a way to compress a big tensor

Represents 2^N amplitudes using only
($2\ N\ m^2$) parameters

Why are MPS useful in physics?

1. Principled

Large enough matrices, represent *any* wavefunction

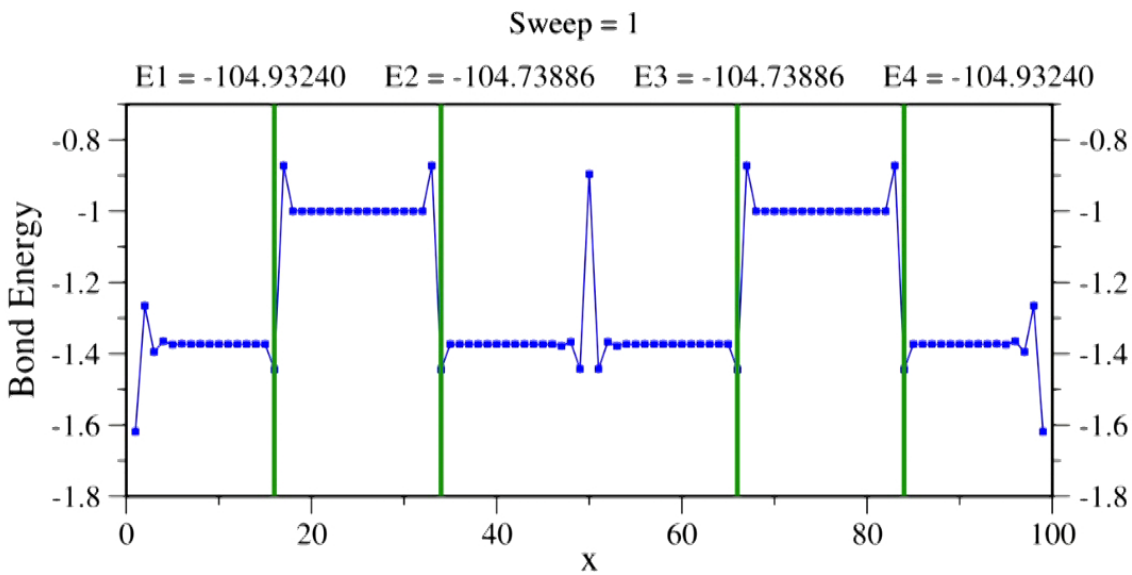
Proof (Hastings, 2007) that one-dimensional systems
with exponentially decaying correlations*
are 'close' to matrix product states

Error decreases rapidly (exponentially in practice)
and *error per site* is constant

*(technically: gap between ground state and excited states)

2. Powerful Algorithms

Matrix product ansatz comes with sophisticated optimization techniques



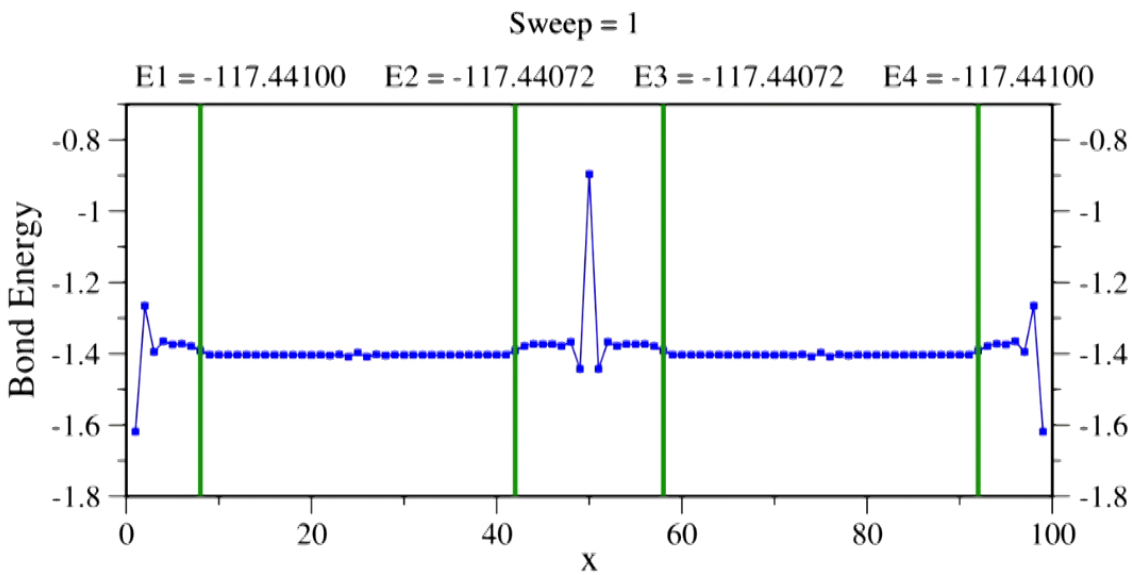
DMRG algorithm

White, PRL **69**, 2863 (1992)

Stoudenmire, White, PRB **87**, 155137 (2013)

2. Powerful Algorithms

Matrix product ansatz comes with sophisticated optimization techniques



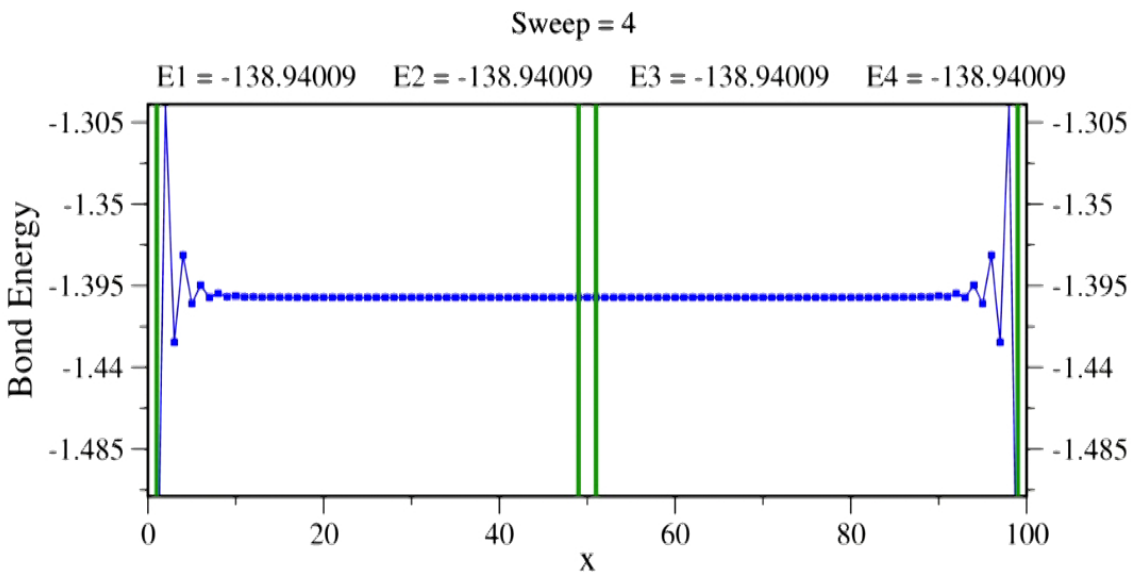
DMRG algorithm

White, PRL **69**, 2863 (1992)

Stoudenmire, White, PRB **87**, 155137 (2013)

2. Powerful Algorithms

Matrix product ansatz comes with sophisticated optimization techniques



DMRG algorithm

White, PRL **69**, 2863 (1992)

Stoudenmire, White, PRB **87**, 155137 (2013)

"In one dimension... it is at the moment the closest to an ultimate weapon as one can dream of."

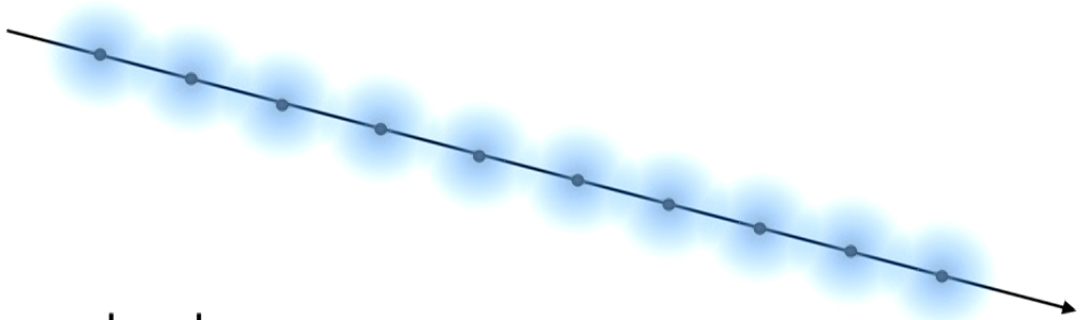
– Thierry Giamarchi

"Quantum Physics in One Dimension"

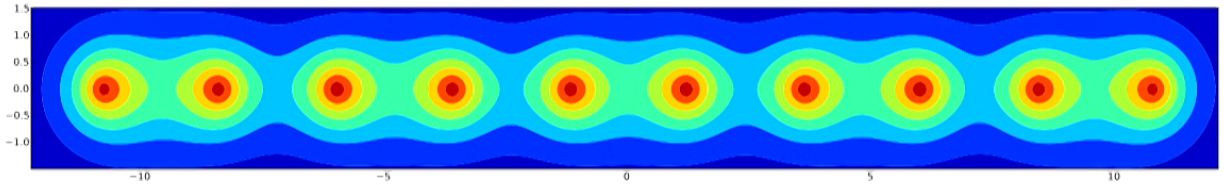
Very often get exact answer

Only takes as many parameters as needed

Application: Quantum chemistry



H_{10} molecule

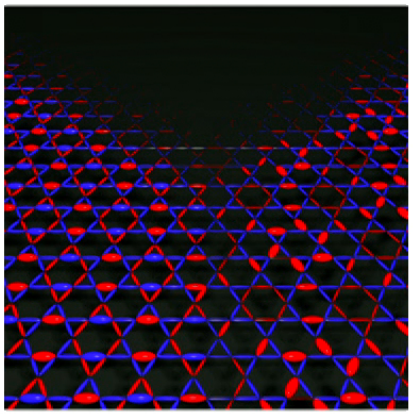


Chan, Sharma, *Annu. Rev. Phys. Chem.* **62,** 465 (2011)
Stoudenmire, White, *Phys. Rev. Lett.* ***119***, *046401 (2017)*

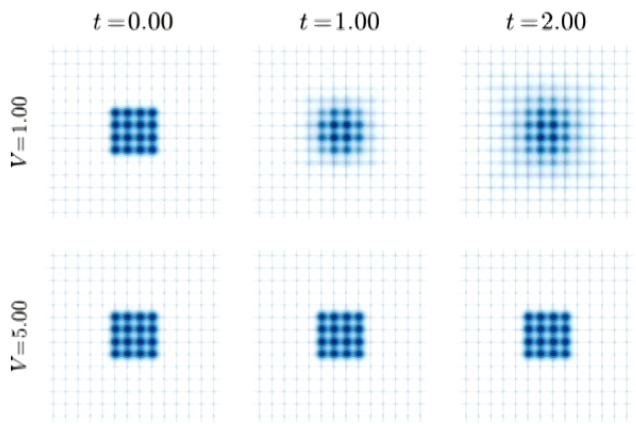
Application: two-dimensional many-body systems

Deal with second dimension by "rasterizing" or snaking matrix product state

Surprisingly powerful approach (state of the art for many systems)



Yan, Huse, White, *Science* **332**, 1173 (2011)



Zaletel, Mong, Karrasch, Moore, Pollmann, *PRB* **91**, 165112 (2015)

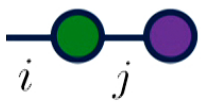
Tensor Diagrams

Helpful to draw N-index tensor as blob with N lines

$$\Psi^{s_1 s_2 s_3 \cdots s_N} =$$

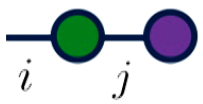
No symmetries or transformation properties assumed

Joining lines implies contraction, can omit names



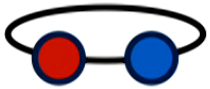
$$\longleftrightarrow \quad \sum_j M_{ij} v_j$$

Joining lines implies contraction, can omit names

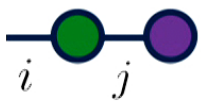


$$\longleftrightarrow \quad \sum_j M_{ij} v_j$$

$$\longleftrightarrow \quad A_{ij} B_{jk} \quad = AB$$

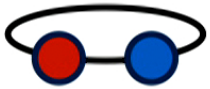


Joining lines implies contraction, can omit names



$$\longleftrightarrow \quad \sum_j M_{ij} v_j$$

$$\longleftrightarrow \quad A_{ij} B_{jk} = AB$$

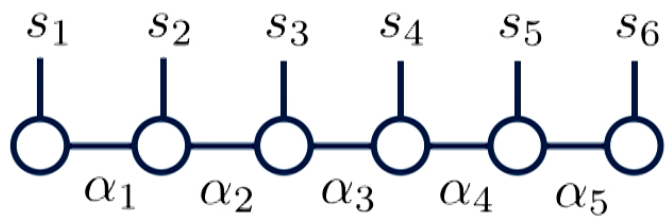


$$\longleftrightarrow \quad A_{ij} B_{ji} = \mathrm{Tr}[AB]$$

Matrix product state in diagram notation

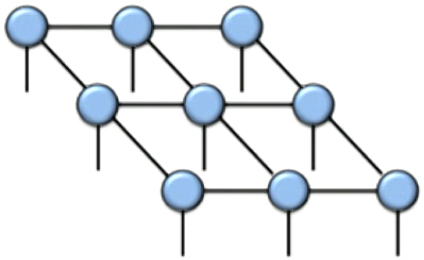
$$\Psi^{s_1 s_2 s_3 s_4 s_5 s_6} = \sum_{\alpha} M^{s_1}_{\alpha_1} M^{s_2}_{\alpha_1 \alpha_2} M^{s_3}_{\alpha_2 \alpha_3} M^{s_4}_{\alpha_3 \alpha_4} M^{s_5}_{\alpha_4 \alpha_5} M^{s_6}_{\alpha_5}$$

$=$



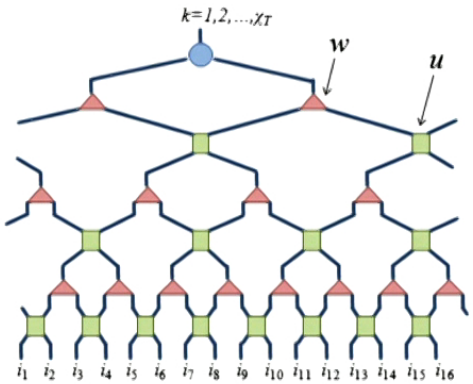
Can suppress index names, very convenient

Besides MPS, other successful tensor are PEPS and MERA



PEPS

(2D systems)



MERA

(critical systems)

Evenbly, Vidal, PRB **79**, 144108 (2009)

Verstraete, Cirac, cond-mat/0407066 (2004)

Orus, Ann. Phys. **349**, 117 (2014)

Tensor Network Machine Learning

Tensor Network Machine Learning

David Schwab

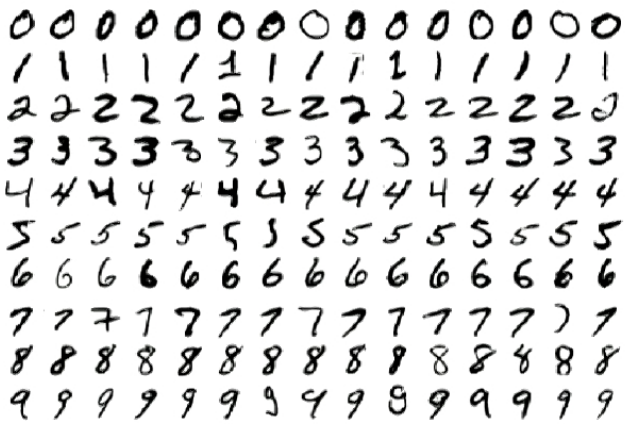
Stoudenmire, Schwab, *Advanced in Neural Information Processing Systems (NIPS)*, **29**, 4799 [arxiv:1605.05775]

Raw data vectors

$$\mathbf{x} = (x_1, x_2, x_3, \ldots, x_N)$$

Example: grayscale images, components of $\mathbf{x}$ are pixels

$$x_j \in [0, 1]$$



Propose following model

$$f(\mathbf{x}) = W \cdot \Phi(\mathbf{x})$$

$$= \sum_{\mathbf{s}} W_{s_1 s_2 s_3 \cdots s_N} \; x_1^{s_1} x_2^{s_2} x_3^{s_3} \cdots x_N^{s_N} \qquad s_j = 0, 1$$

Weights are N-index tensor
Like N-site wavefunction

Novikov, Trofimov, Oseledets, arxiv:1605.03795

N=3 example:

$$f(\mathbf{x}) = W \cdot \Phi(\mathbf{x}) = \sum_{\mathbf{s}} W_{s_1 s_2 s_3}\; x_1^{s_1} x_2^{s_2} x_3^{s_3}$$

$$= W_{000} + W_{100}\, x_1 + W_{010}\, x_2 + W_{001}\, x_3$$

$$+ W_{110}\, x_1 x_2 + W_{101}\, x_1 x_3 + W_{011}\, x_2 x_3$$

$$+ W_{111}\, x_1 x_2 x_3$$

Contains linear classifier, plus other "feature maps"

More generally, apply local "feature maps" $\phi^{s_j}(x_j)$

$$f(\mathbf{x}) = W \cdot \Phi(\mathbf{x})$$

$$= \sum_{\mathbf{s}} W_{s_1 s_2 s_3 \cdots s_N} \phi^{s_1}(x_1)\phi^{s_2}(x_2)\phi^{s_3}(x_3)\cdots\phi^{s_N}(x_N)$$

Highly expressive!

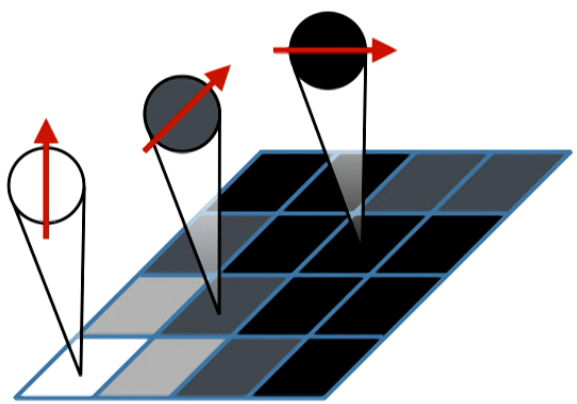
Cohen et al. arxiv:1509.05009
Novikov, Trofimov, Oseledets, arxiv:1605.03795
Stoudenmire, Schwab, arxiv:1605.05775

$\mathbf{x} =$ input

For example, following local feature map

$$\phi(x_j) = \left[\cos\left(\frac{\pi}{2}x_j\right), \sin\left(\frac{\pi}{2}x_j\right)\right] \qquad x_j \in [0,1]$$

Picturesque idea of pixels as "spins" or "qubits"



$\mathbf{x} =$ input

$\phi =$ local feature map

Total feature map $\Phi(\mathbf{x})$

$$\Phi^{s_1 s_2 \cdots s_N}(\mathbf{x}) = \phi^{s_1}(x_1) \otimes \phi^{s_2}(x_2) \otimes \cdots \otimes \phi^{s_N}(x_N)$$

- Tensor product of local feature maps / vectors
- Just like product state wavefunction of spins
- Vector in 2^N dimensional space

$\mathbf{x} =$ input

$\phi =$ local feature map

Total feature map $\Phi(\mathbf{x})$

More detailed notation

$$\mathbf{x} = [x_1, \quad x_2, \quad x_3, \quad \ldots \quad , \quad x_N]$$

raw inputs

$$\Phi(\mathbf{x}) = \begin{bmatrix} \phi_1(x_1) \\ \phi_2(x_1) \end{bmatrix} \otimes \begin{bmatrix} \phi_1(x_2) \\ \phi_2(x_2) \end{bmatrix} \otimes \begin{bmatrix} \phi_1(x_3) \\ \phi_2(x_3) \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} \phi_1(x_N) \\ \phi_2(x_N) \end{bmatrix}$$

feature vector

$\mathbf{x} =$ input

$\phi =$ local feature map

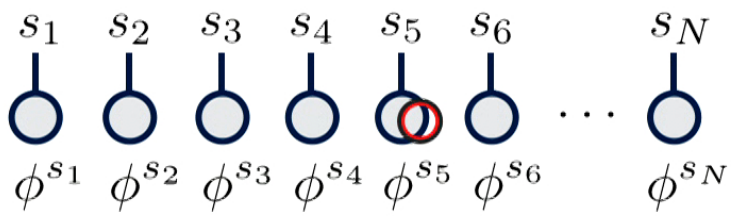
Total feature map $\Phi(\mathbf{x})$

Tensor diagram notation

$$\mathbf{x} = [x_1, \quad x_2, \quad x_3, \quad \ldots \quad , \quad x_N]$$

raw inputs

$$\Phi(\mathbf{x}) =$$



feature vector

Construct decision function

$$f(\mathbf{x}) = W \cdot \Phi(\mathbf{x})$$

Main approximation

$W =$

order-N tensor

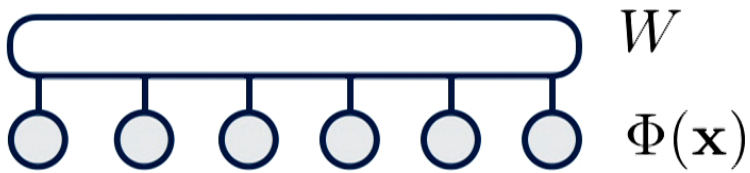
$\approx$

matrix product state (MPS)

Construct decision function

$$\boxed{f(\mathbf{x}) = W \cdot \Phi(\mathbf{x})}$$

$f(\mathbf{x}) =$



Main approximation

$W =$

order-N tensor

$\approx$

matrix product state (MPS)

Can use algorithm similar to DMRG to optimize

Number of parameters is *adaptive*

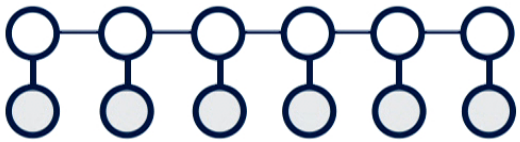
Scaling is $N \cdot N_T \cdot m^3$

$N =$ size of input

$N_T =$ size of training set

$m =$ MPS bond dimension

$f(\mathbf{x}) =$



W

$\Phi(\mathbf{x})$

Can use algorithm similar to DMRG to optimize

Number of parameters is *adaptive*

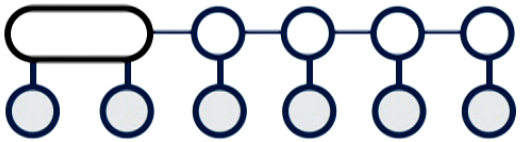
Scaling is $N \cdot N_T \cdot m^3$

$N =$ size of input

$N_T =$ size of training set

$m =$ MPS bond dimension

$f(\mathbf{x}) =$



W

$\Phi(\mathbf{x})$

Can use algorithm similar to DMRG to optimize

Number of parameters is *adaptive*

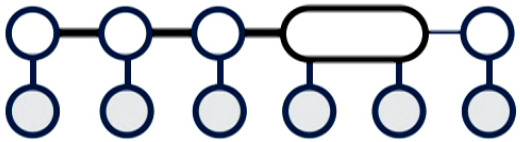
Scaling is $N \cdot N_T \cdot m^3$

$N =$ size of input

$N_T =$ size of training set

$m =$ MPS bond dimension

$f(\mathbf{x}) =$



W

$\Phi(\mathbf{x})$

Why should this work at all?

Linear classifier $f(\mathbf{x}) = V \cdot \mathbf{x}$ exactly m=2 MPS

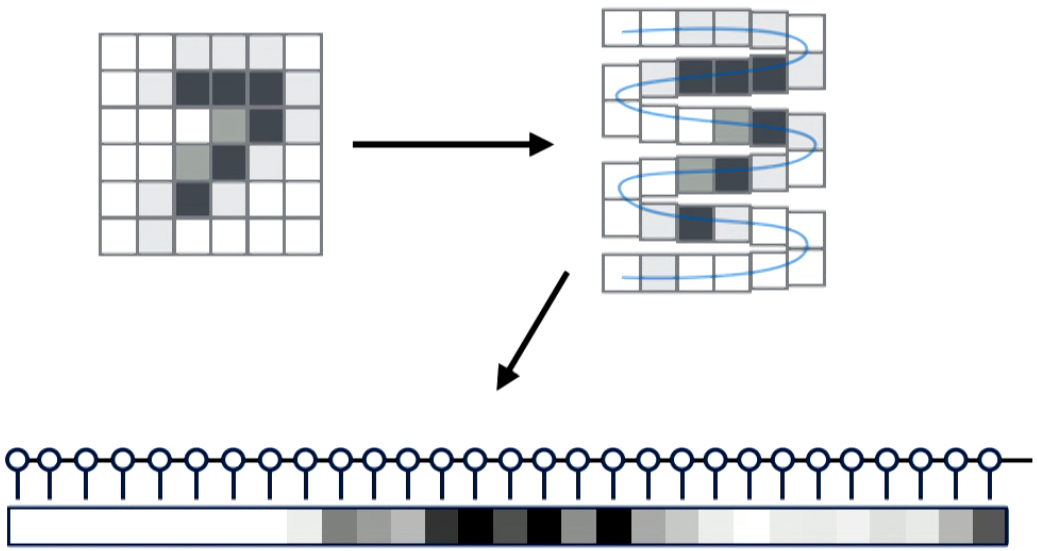
$$W = \begin{bmatrix} V_0 & 1 \end{bmatrix} \begin{bmatrix} \hat{1} & 0 \\ \hat{V}_1 & \hat{1} \end{bmatrix} \begin{bmatrix} \hat{1} & 0 \\ \hat{V}_2 & \hat{1} \end{bmatrix} \begin{bmatrix} \hat{1} & 0 \\ \hat{V}_3 & \hat{1} \end{bmatrix} \cdots$$

$$\hat{1} = [1 \;\; 0] \qquad f(\mathbf{x}) = W \cdot \Phi(\mathbf{x})$$

$$\hat{V}_j = [0 \;\; V_j] \qquad \phi^{s_j}(x_j) = [1\,,\, x_j]$$

Novikov, Trofimov, Oseledets, arxiv:1605.03795

Experiment: handwriting classification (MNIST)



Train to 99.95% accuracy on 60,000 training images

Obtain **99.03%** accuracy on 10,000 test images
(only 97 incorrect)

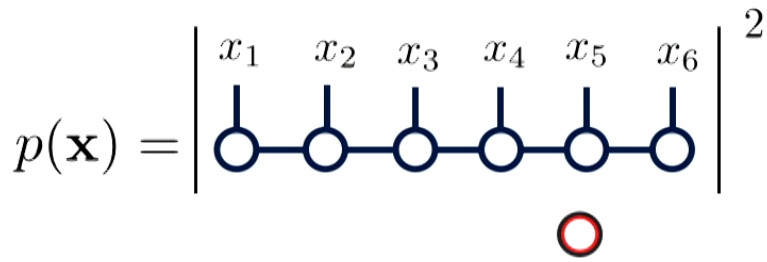
Stoudenmire, Schwab, arxiv:1605.05775

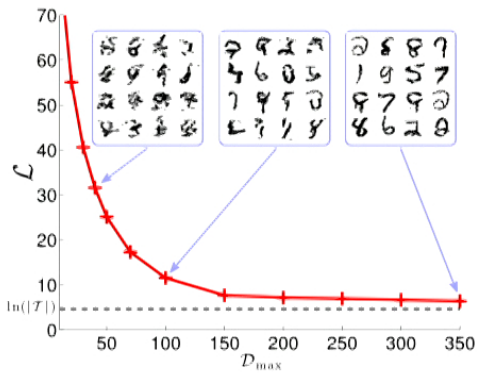
Tensor Network Machine Learning Studies

Unsupervised Generative Modeling Using MPS

Zhao-Yu Han, Jun Wang, Heng Fan, Lei Wang, Pan Zhang *(arxiv:1709.01662)*

- Map data to product state, tensor network weights
- Squared output is probability – "Born machine"
- "Perfect" sampling (no autocorrelation)





Negative Log-Likelihood

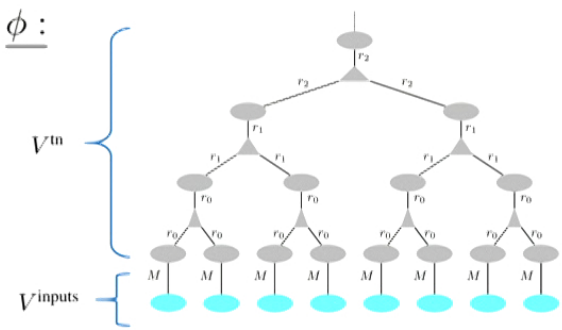
Reconstructing Testing Images

Tensor Network Machine Learning Studies

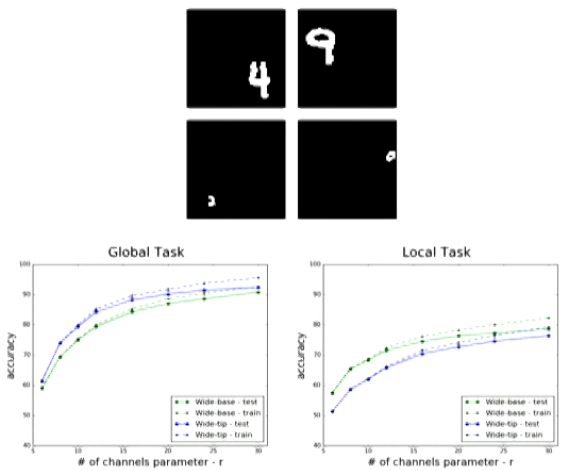
Deep Learning and Quantum Entanglement...

Yoav Levine, David Yakira, Nadav Cohen, Amnon Shashua (arxiv:1704.01552)

- "ConvAC" deep neural net = tree tensor network
- Tensor network rank as capacity of model
- Experiment on "inductive bias" of model architecture



Tree Network as a Deep Neural Net

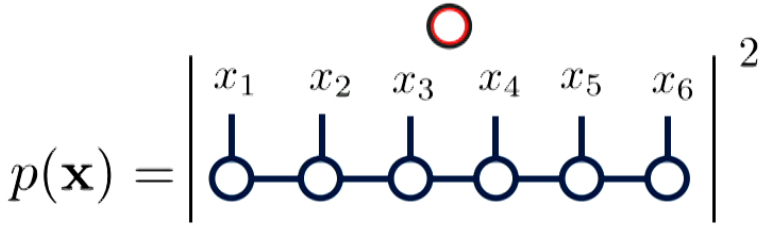


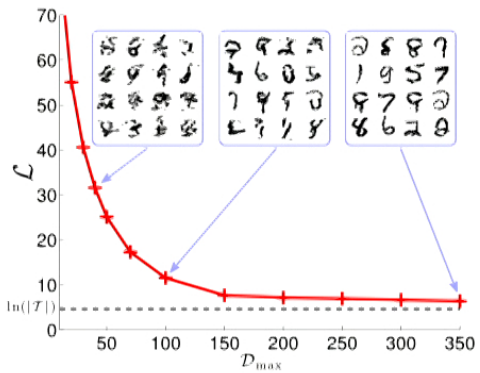
Inductive Bias Experiment

Unsupervised Generative Modeling Using MPS

Zhao-Yu Han, Jun Wang, Heng Fan, Lei Wang, Pan Zhang *(arxiv:1709.01662)*

- Map data to product state, tensor network weights
- Squared output is probability – "Born machine"
- "Perfect" sampling (no autocorrelation)





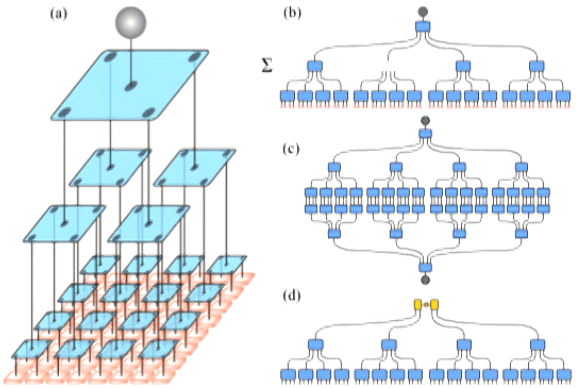
Negative Log-Likelihood

Reconstructing Testing Images

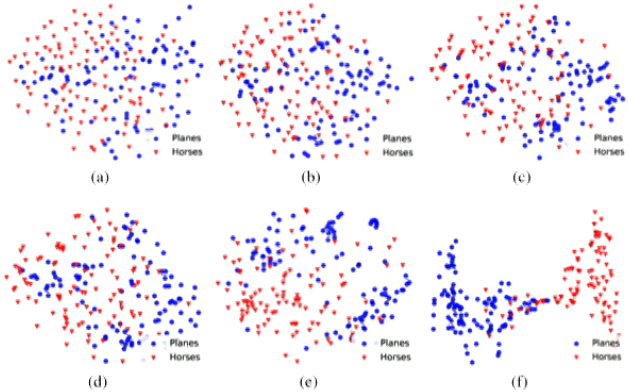
Machine Learning By Hierarchical Tensor Networks...

Ding Liu, Shi-Ju Ran, Peter Wittek, Cheng Peng, Raul Blazquez Garcia, Gang Su, Maciej Lewenstein (arxiv:1710.04833)

- Supervised learning with tree tensor networks
- Tests on MNIST, CIFAR-10
- Studied properties of the trained model (feature representations, entanglement)



Model Architecture

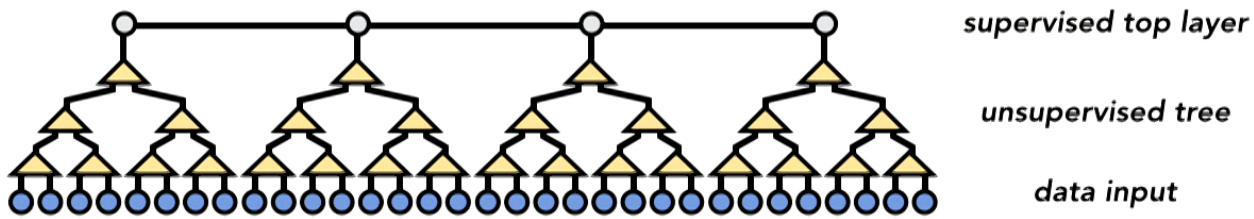


Data Representation at Different Scales

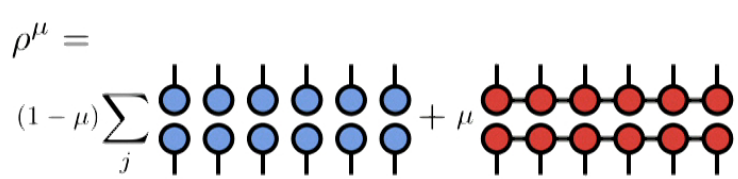
Learning Relevant Features of Data...

E.M. Stoudenmire (Quant. Sci. Tech. 3, 034003 [2018])

- Unsupervised determination of tree tensor network (compress data)
- Supervised training of top layer
- Excellent performance with "features" determined by tree tensors

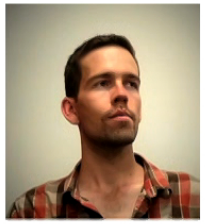


89% accuracy on Fashion MNIST data set



mixed training supervised / unsupervised

Quantum Machine Learning with Tensor Networks



Bill Huggins

Huggins, Patil, Whaley, Stoudenmire, arxiv:1803.11537

Grant, Benedetti, et al., arxiv:1804.03680

Two recent proposals for machine learning with a quantum computer

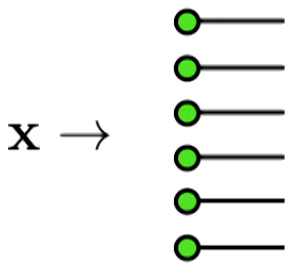
1. supervised / discriminative

Farhi, Neven, arxiv:1802.0600

Schuld, Killoran, arxiv:1803.07128

Two recent proposals for machine learning with a quantum computer

1. supervised / discriminative

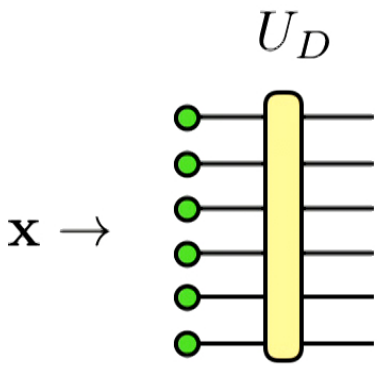


Farhi, Neven, arxiv:1802.0600

Schuld, Killoran, arxiv:1803.07128

Two recent proposals for machine learning with a quantum computer

1. supervised / discriminative

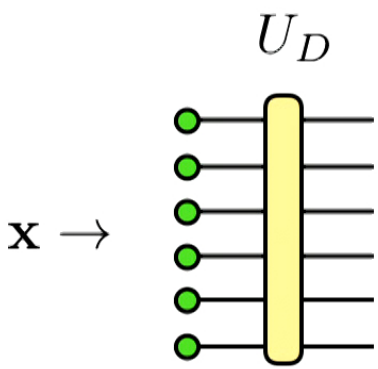


Farhi, Neven, arxiv:1802.0600

Schuld, Killoran, arxiv:1803.07128

Two recent proposals for machine learning with a quantum computer

1. supervised / discriminative

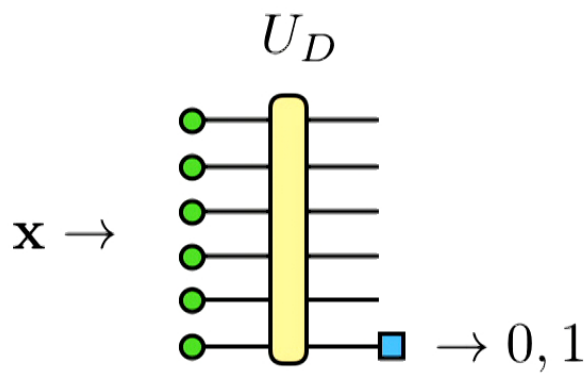


Farhi, Neven, arxiv:1802.0600

Schuld, Killoran, arxiv:1803.07128

Two recent proposals for machine learning with a quantum computer

1. supervised / discriminative



Farhi, Neven, arxiv:1802.0600

Schuld, Killoran, arxiv:1803.07128

Two recent proposals for machine learning with a quantum computer

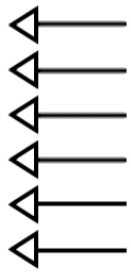
2. unsupervised / generative

Gao, Zhang, Duan, arxiv:1711.02038

Benedetti, Garcia-Pintos, Nam, Perdomo-Ortiz, arxiv:1801.07686

Two recent proposals for machine learning with a quantum computer

2. unsupervised / generative

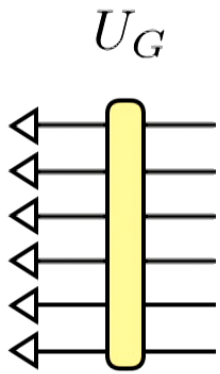


Gao, Zhang, Duan, arxiv:1711.02038

Benedetti, Garcia-Pintos, Nam, Perdomo-Ortiz, arxiv:1801.07686

Two recent proposals for machine learning with a quantum computer

2. unsupervised / generative

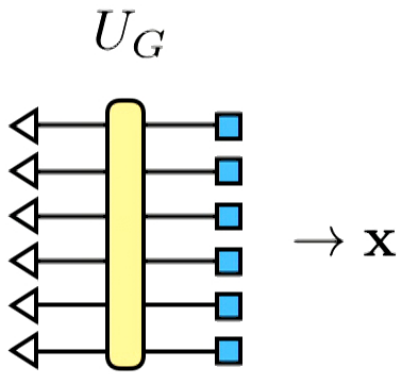


Gao, Zhang, Duan, arxiv:1711.02038

Benedetti, Garcia-Pintos, Nam, Perdomo-Ortiz, arxiv:1801.07686

Two recent proposals for machine learning with a quantum computer

2. unsupervised / generative

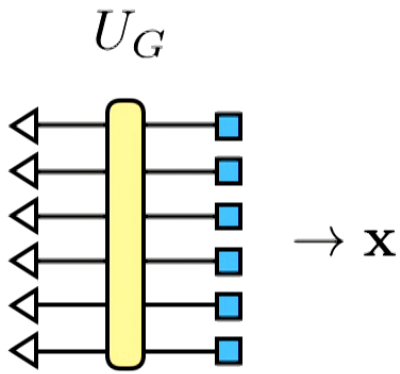


Gao, Zhang, Duan, arxiv:1711.02038

Benedetti, Garcia-Pintos, Nam, Perdomo-Ortiz, arxiv:1801.07686

Two recent proposals for machine learning with a quantum computer

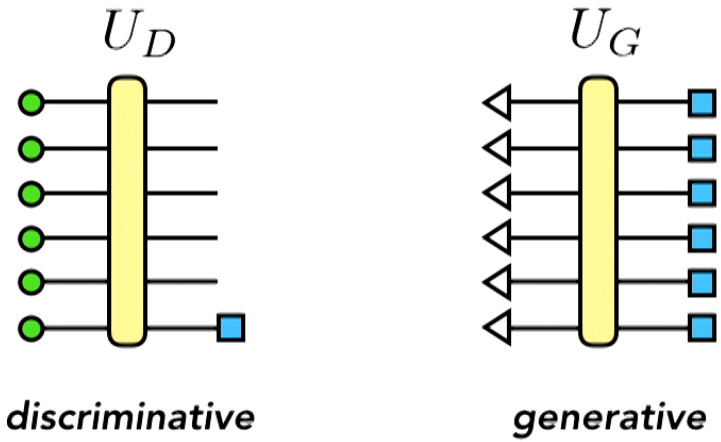
2. unsupervised / generative



Gao, Zhang, Duan, arxiv:1711.02038

Benedetti, Garcia-Pintos, Nam, Perdomo-Ortiz, arxiv:1801.07686

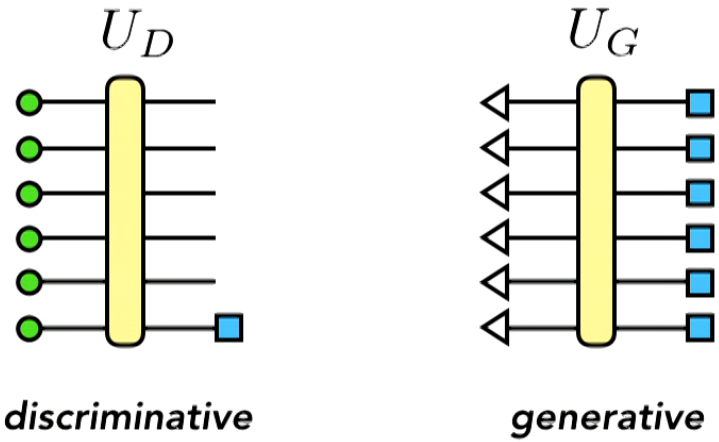
Two issues with these proposals



1. efficient parameterization of N-qubit circuit?
   (vanishing gradient?*)

* McClean, Boixo, et al., arxiv:1803.11173

Two issues with these proposals

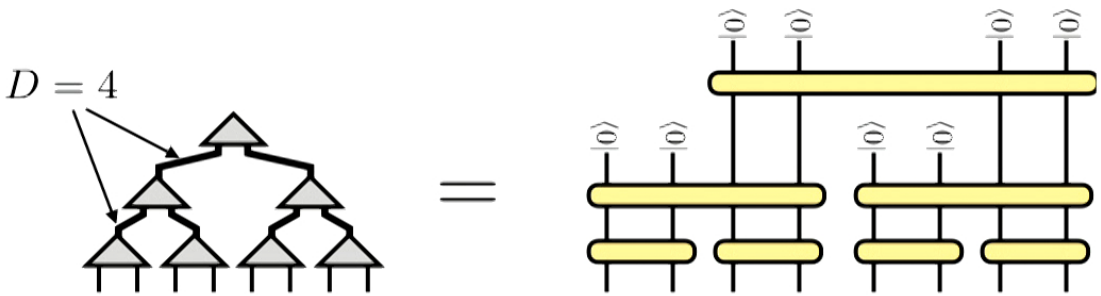


1. efficient parameterization of N-qubit circuit? (vanishing gradient?*)

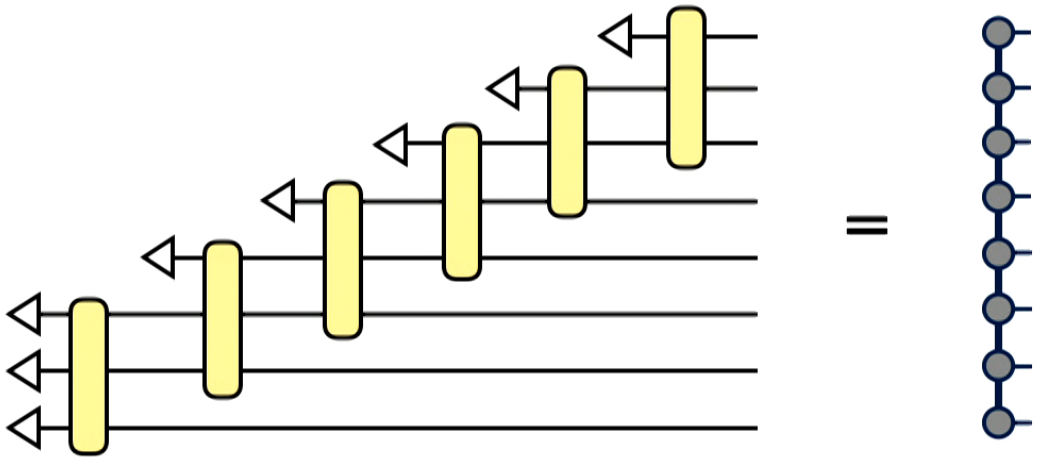
2. require too many qubits for realistic data sizes

* McClean, Boixo, et al., arxiv:1803.11173

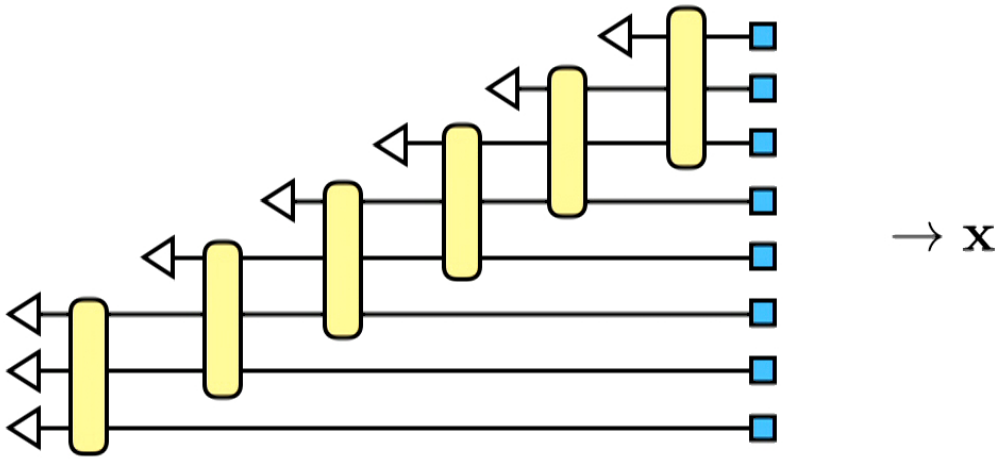
Tensor networks are equivalent to quantum circuits



Quantum circuit for matrix product state (m = 4)

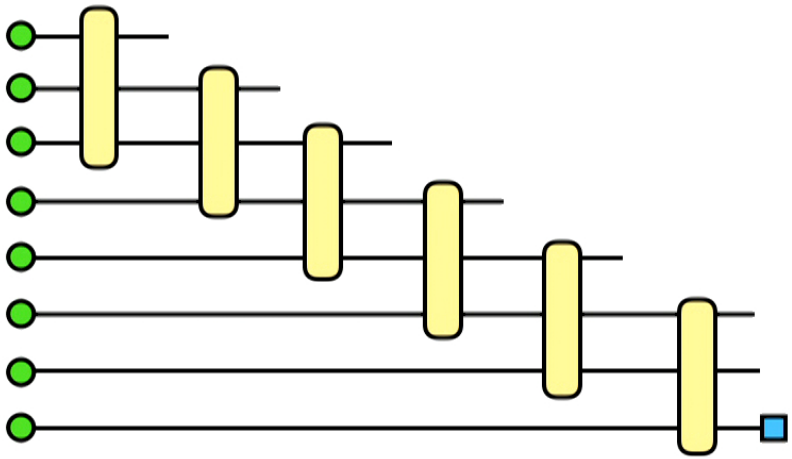


Suggests MPS parameterization of generative quantum model



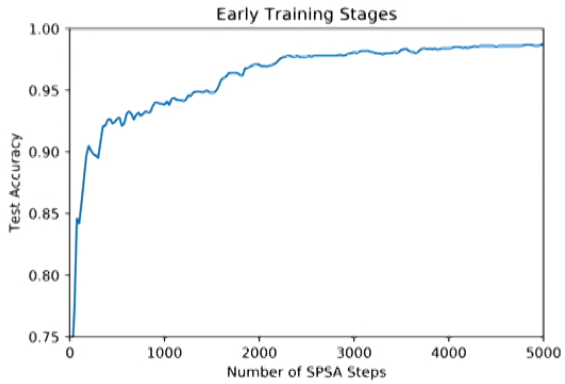
- much fewer paramaters than arbitrary circuit
- can initialize with classically optimized MPS

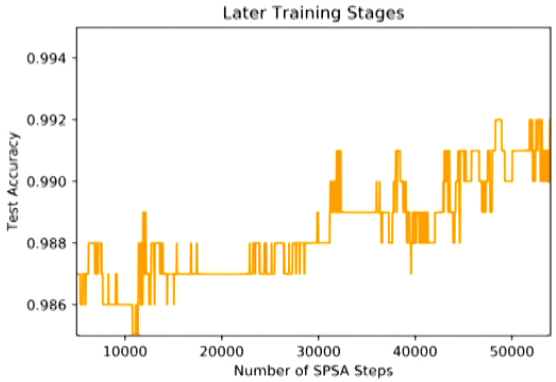
Discriminative model basically the reverse



Test discriminative idea, using only operations available to quantum hardware:

Bill Huggins



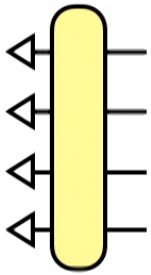


8x8 images (MNIST)
distinguish 0's from 1's

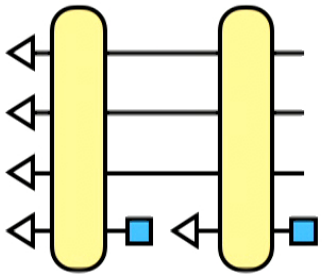
Obtain 99% accuracy
training & test

Sampling higher-dimensional output than number of qubits

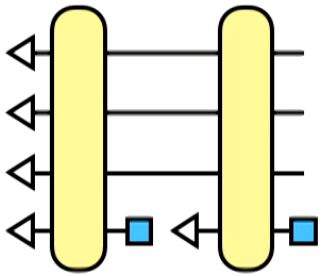
Sampling higher-dimensional output than number of qubits



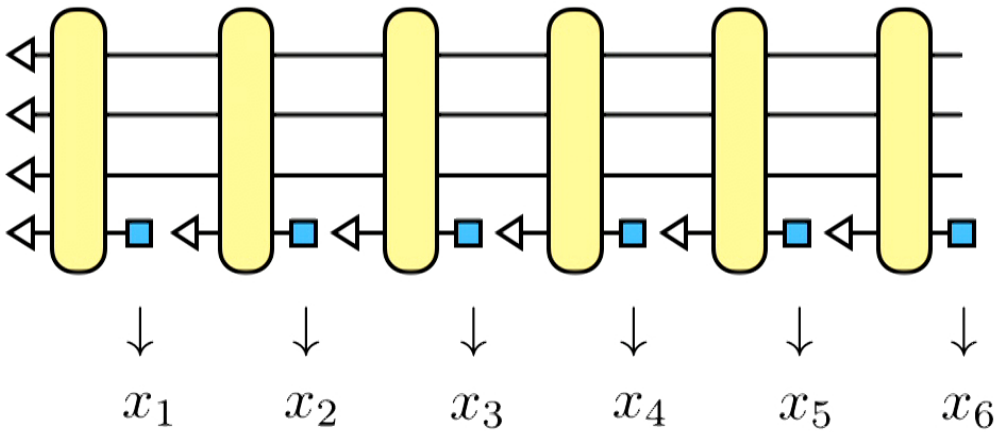
Sampling higher-dimensional output than number of qubits



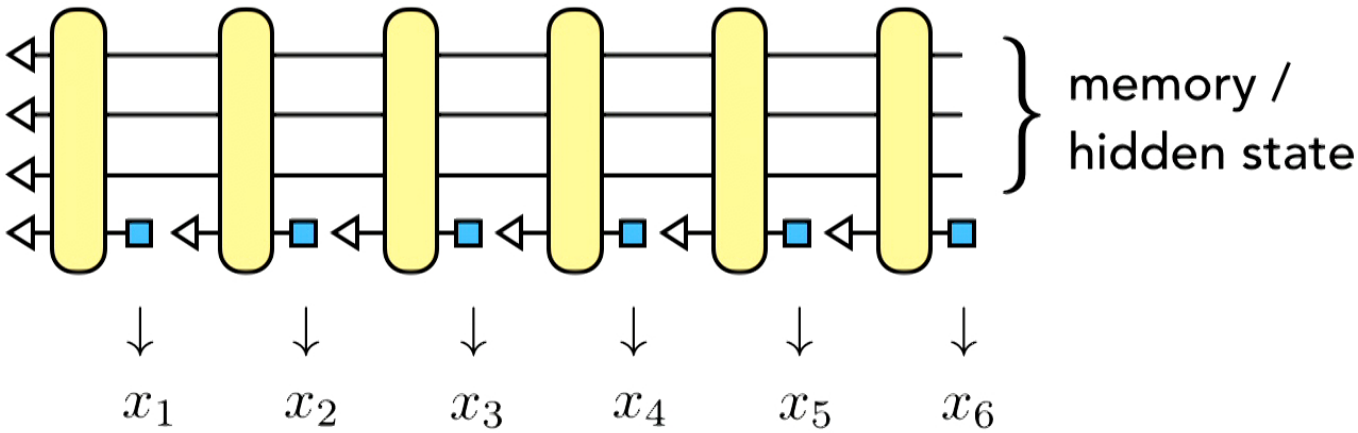
Sampling higher-dimensional output than number of qubits



Sampling higher-dimensional output than number of qubits

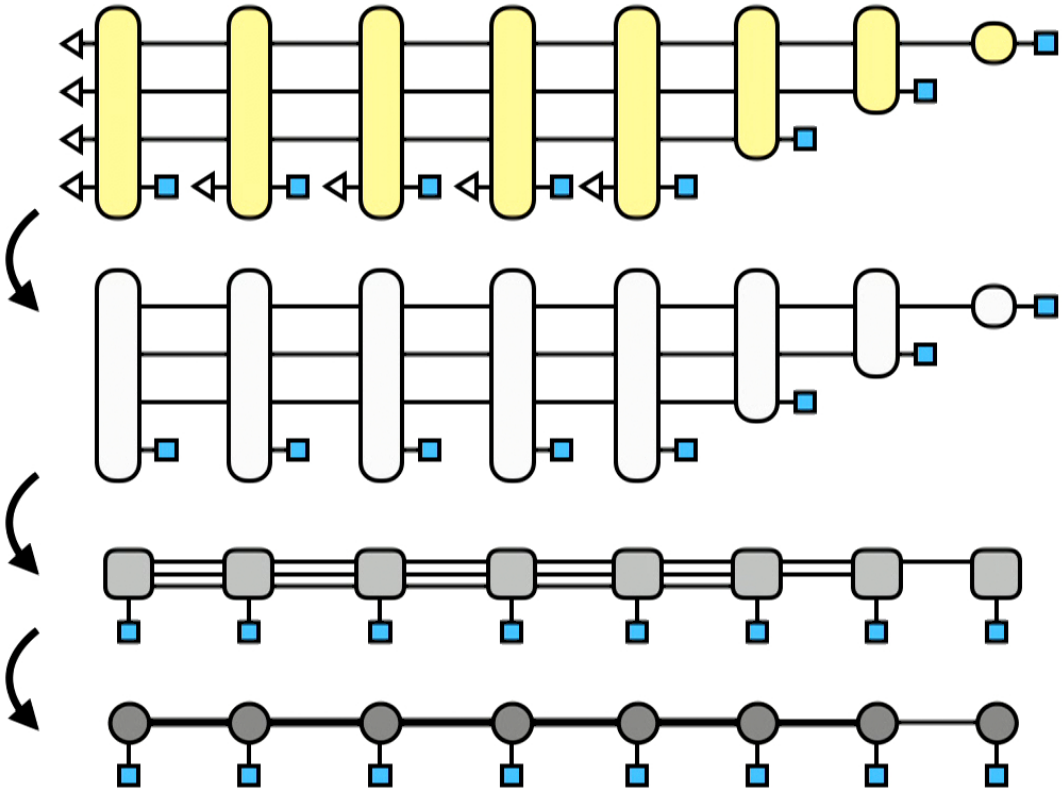


Sampling higher-dimensional output than number of qubits



memory / hidden state size
exponential in number of qubits

Equivalent to sampling from a matrix product state



Conclusions & Future Directions

- Quantum-inspired tensor networks an intriguing alternative to traditional machine learning models
- Better scaling, interesting algorithms, opportunities for theoretical insights
- Continue pushing interpretability, algorithms
- Framework for machine learning with quantum computing