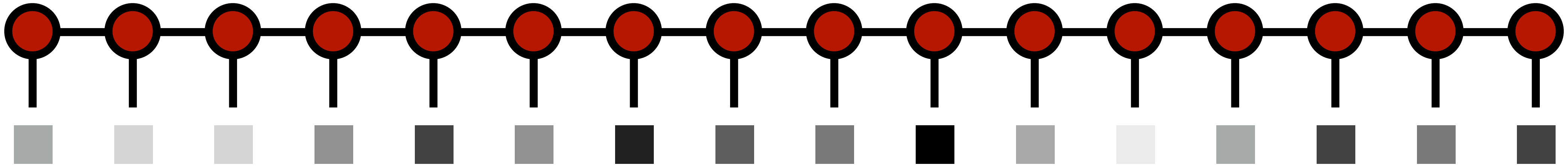
Introduction to Tensor Networks for Machine Learning



E.M. Stoudenmire

Feb 2021 - Benasque SCS

FLATIRON INSTITUTE
SIMONS FOUNDATION

Plan of Lectures

Tensor networks are a natural way to parameterize interesting and powerful machine learning models

Today:

- Intro to Machine Learning
- Intro to Tensor Networks
- Tensor Network Machine Learning

Thursday:

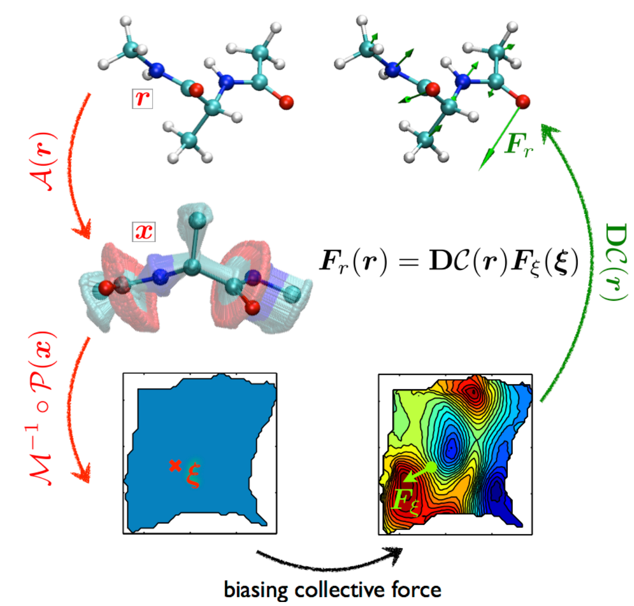
- Architectures beyond MPS
- Potential of new algorithms

Machine learning galvanizing industry & science

Language Processing

Self-driving cars

Medicine



Materials Science / Chemistry

Google rebranded a "machine learning first company"

STEVEN LEVY 06.22.16 12:00 AM

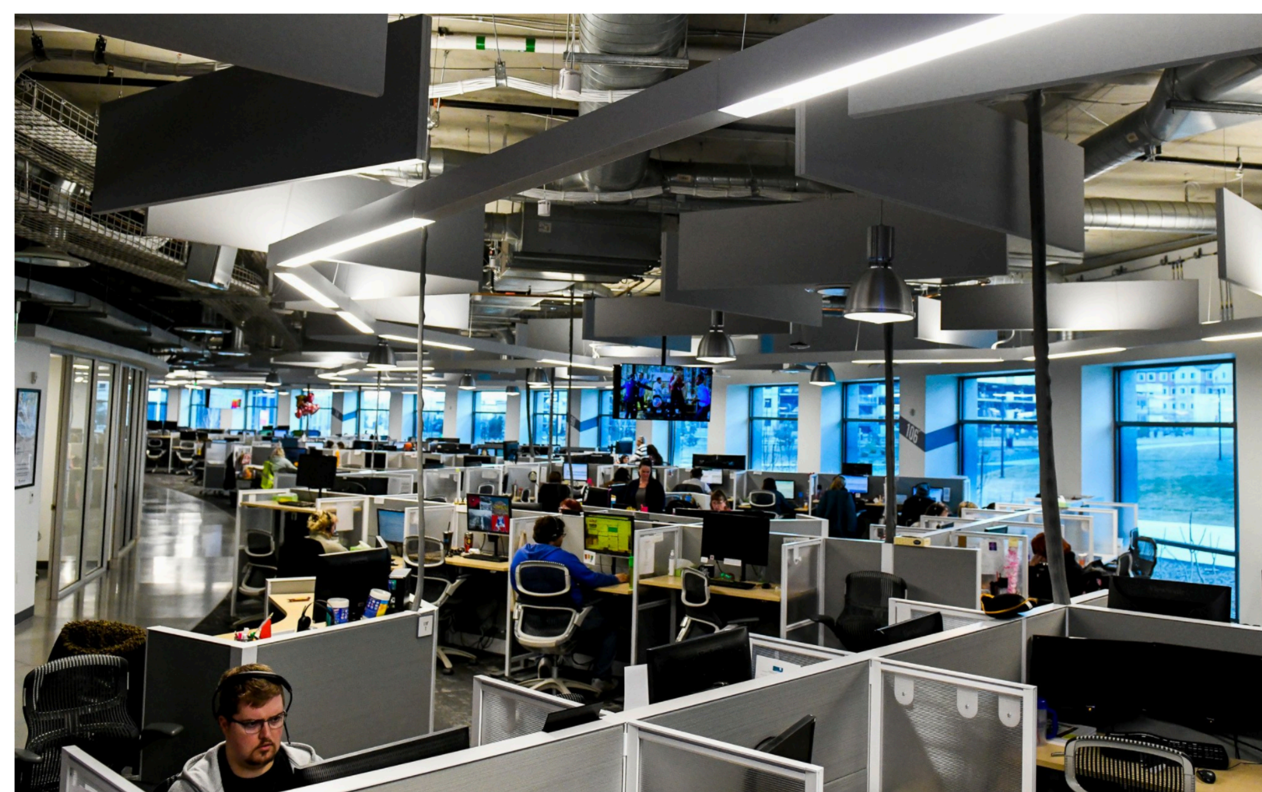
How Google is Remaking Itself as a "Machine Learning First" Company

Neural nets replace linguistic approach to Google Translate

TECHNOLOGY

Why A.I. Researchers at Google Got Desks Next to the Boss

By CADE METZ FEB. 19, 2018



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

Examples of Machine Learning

Image recognition

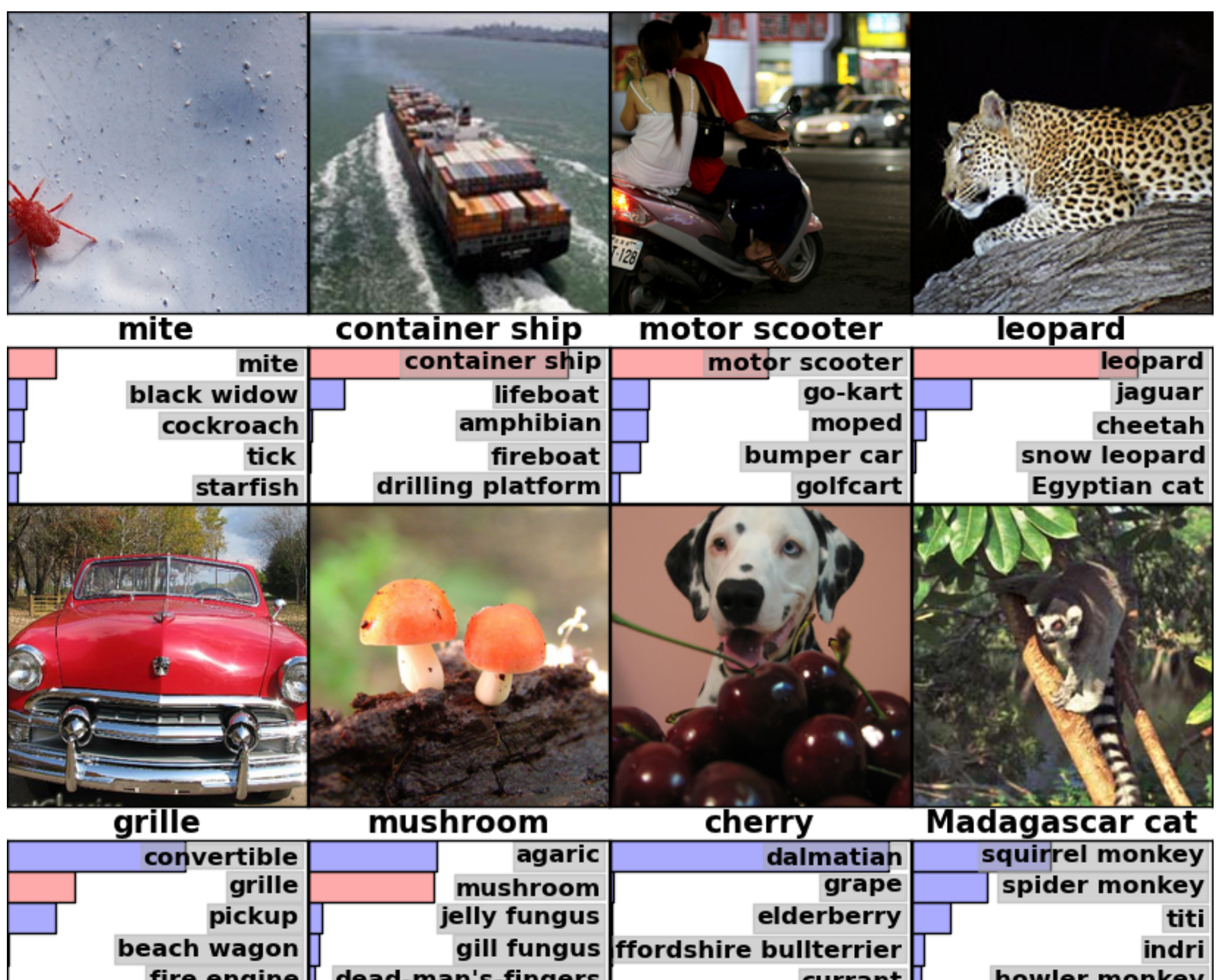
ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

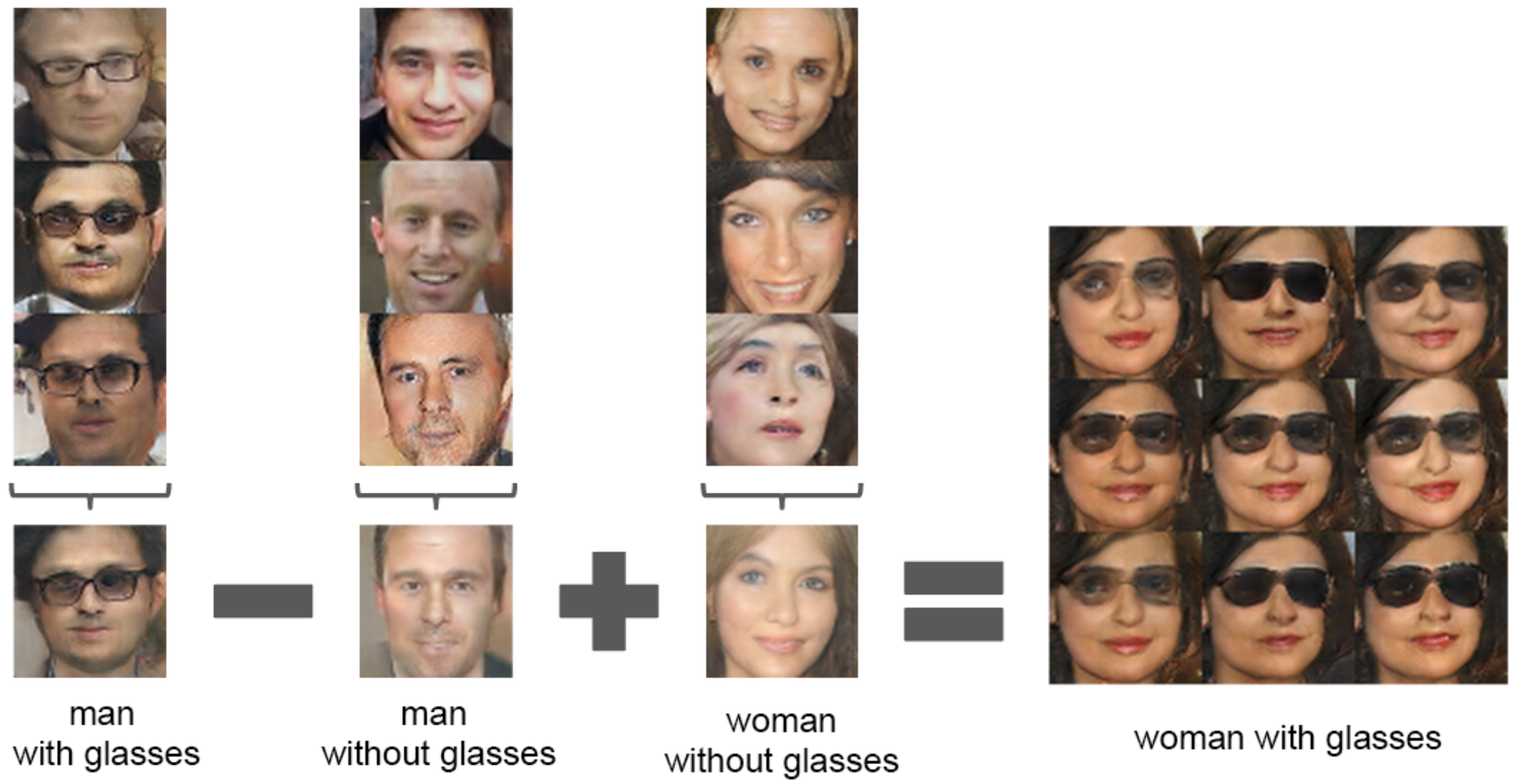
2012 paper that launched recent deep learning era (20k citations)



ImageNet:

- 1.2 million training images (150k test)
- 1000 categories
- 15% neural net error
- 26% next best error

Image Generation

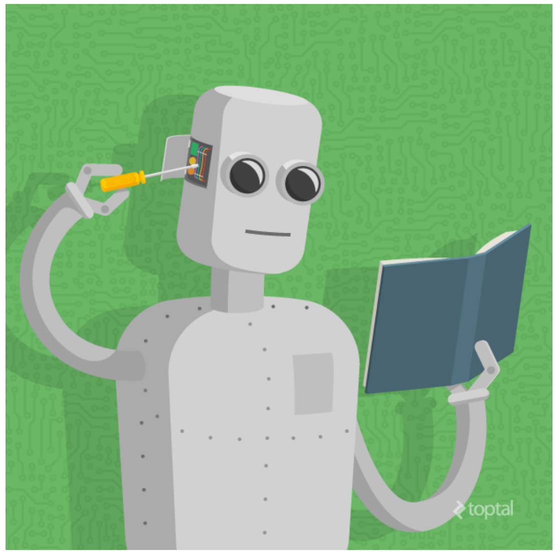


UNSUPERVISED REPRESENTATION LEARNING
WITH DEEP CONVOLUTIONAL
GENERATIVE ADVERSARIAL NETWORKS

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

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

What is machine learning?



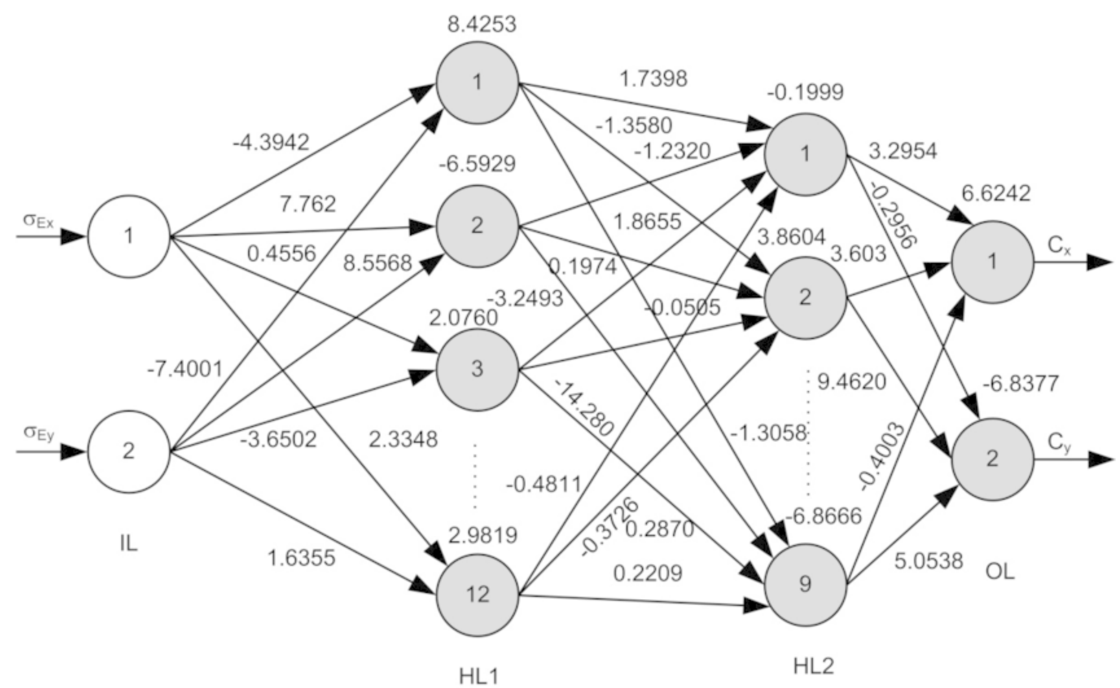
Data driven problem solving

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

Framework / philosophy, not single method

Software 1.0

Software 2.0



Andrej Karpathy Follow

Director of AI at Tesla. Previously Research Scientist at OpenAI and PhD student at Stanford. I like to train deep neural nets on large datasets.

Nov 11, 2017 · 7 min read

https://medium.com/@karpathy/software-2-0-a64152b37c35

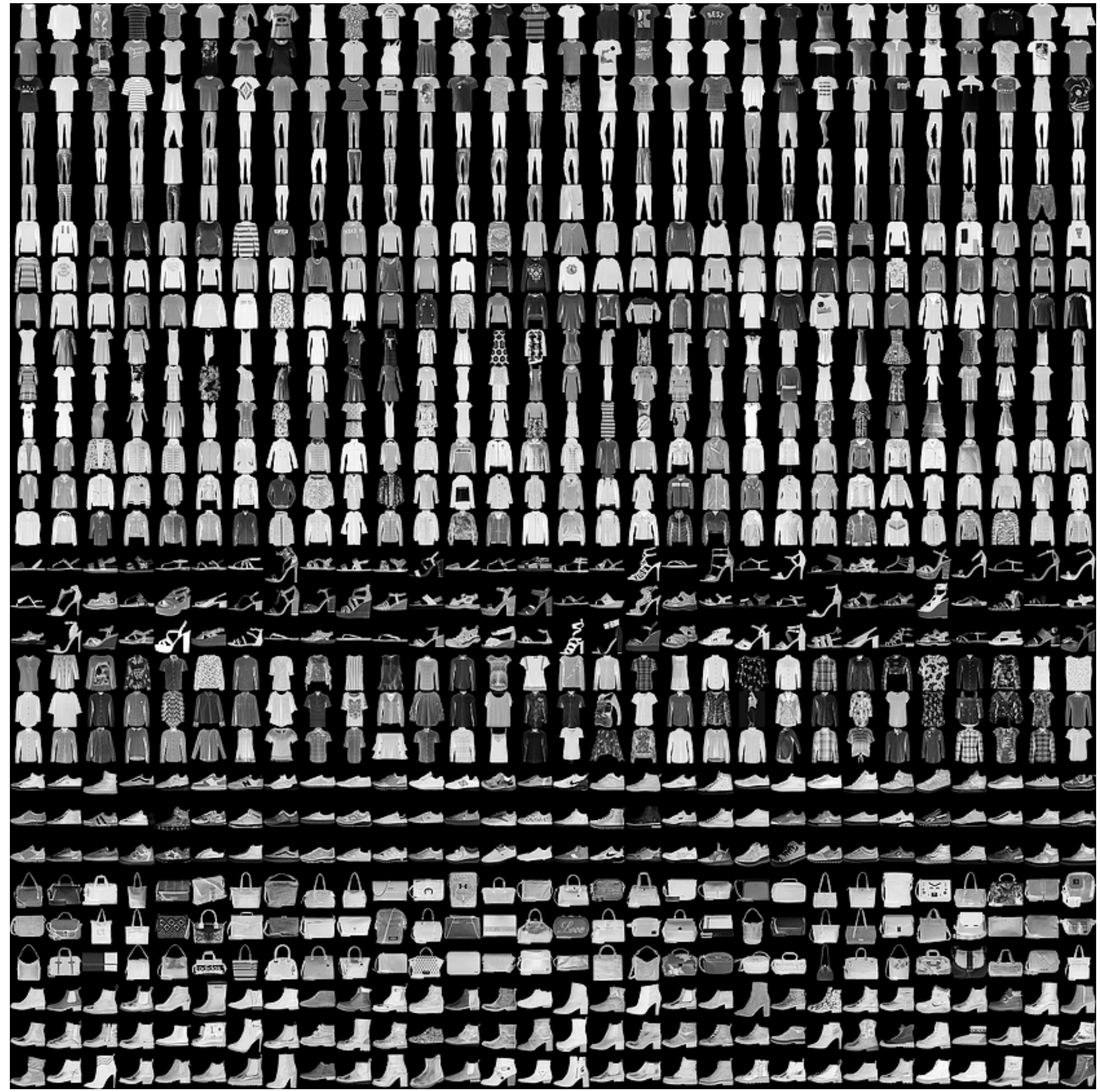
Basics of Machine Learning

Example of a Dataset – Fashion MNIST

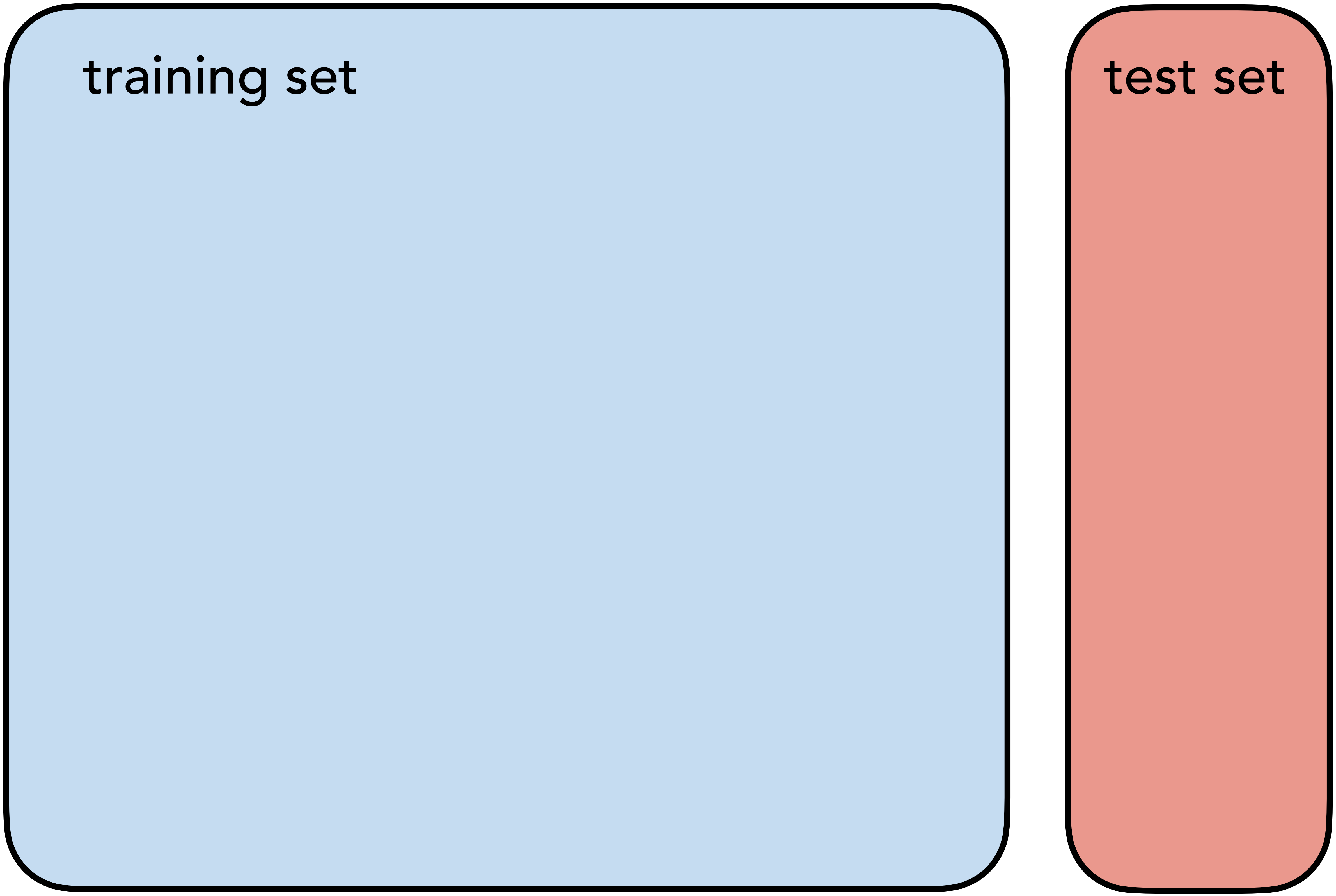
10 categories (labels)

28x28 grayscale

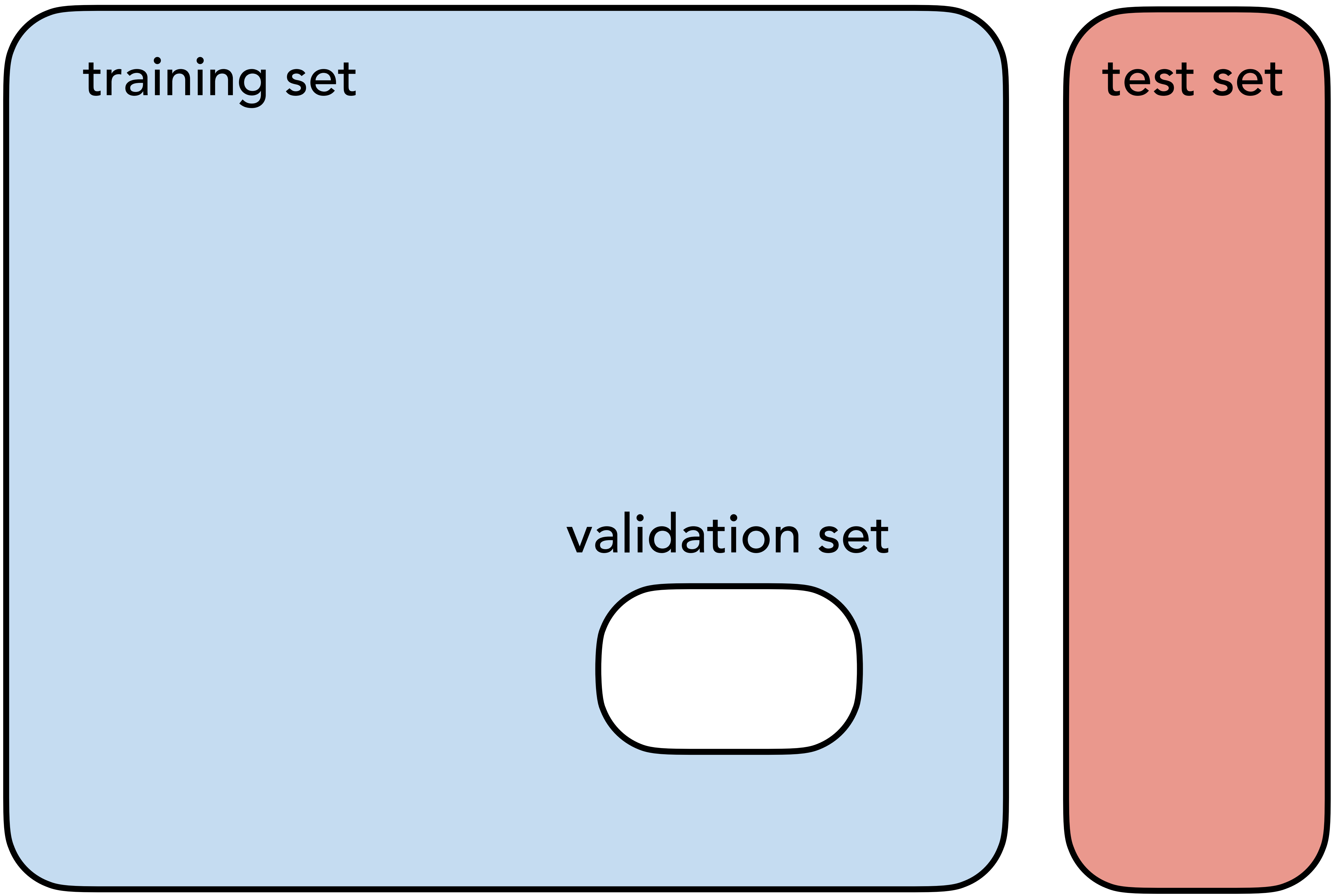
70000 labeled images



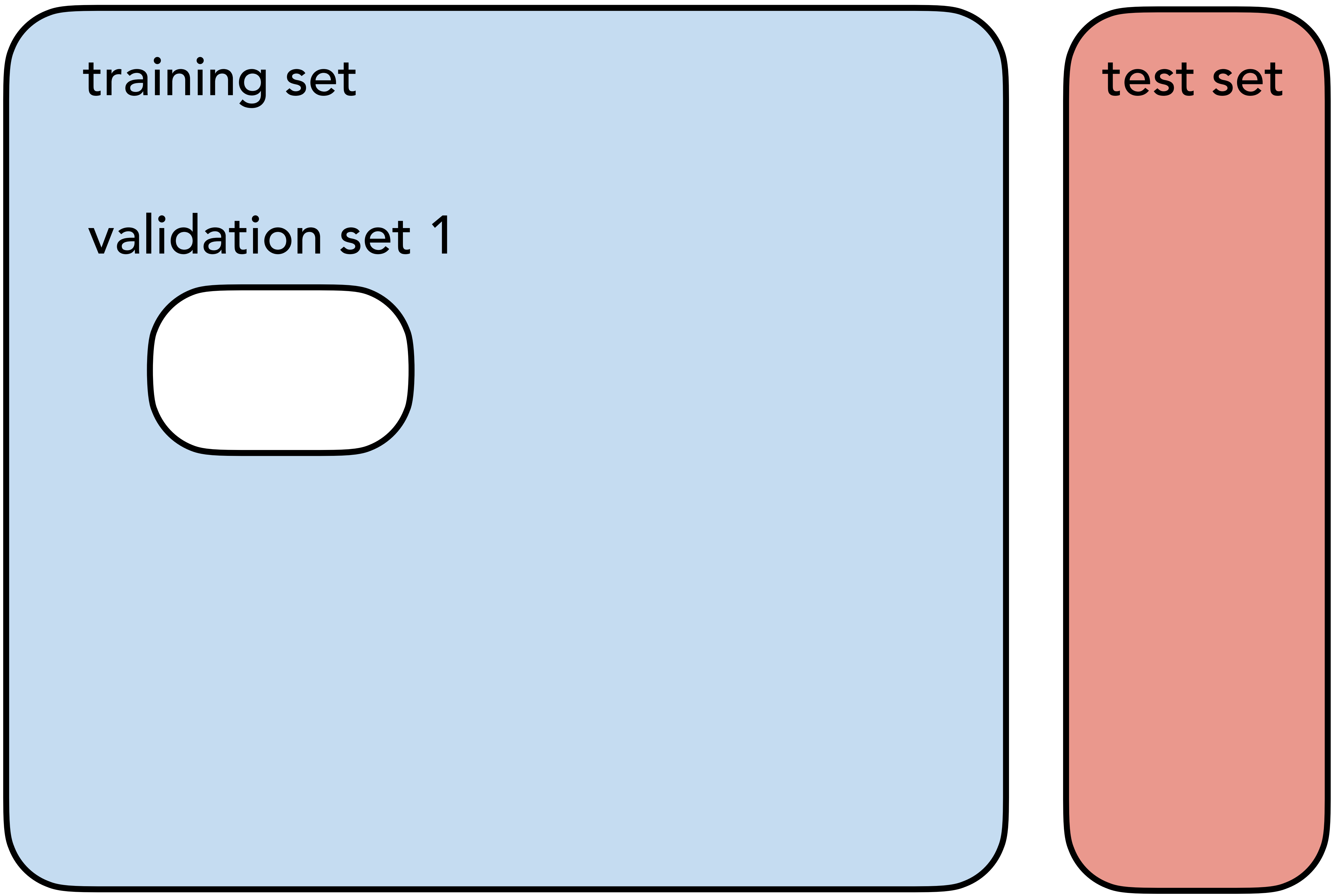
Anatomy of a Dataset



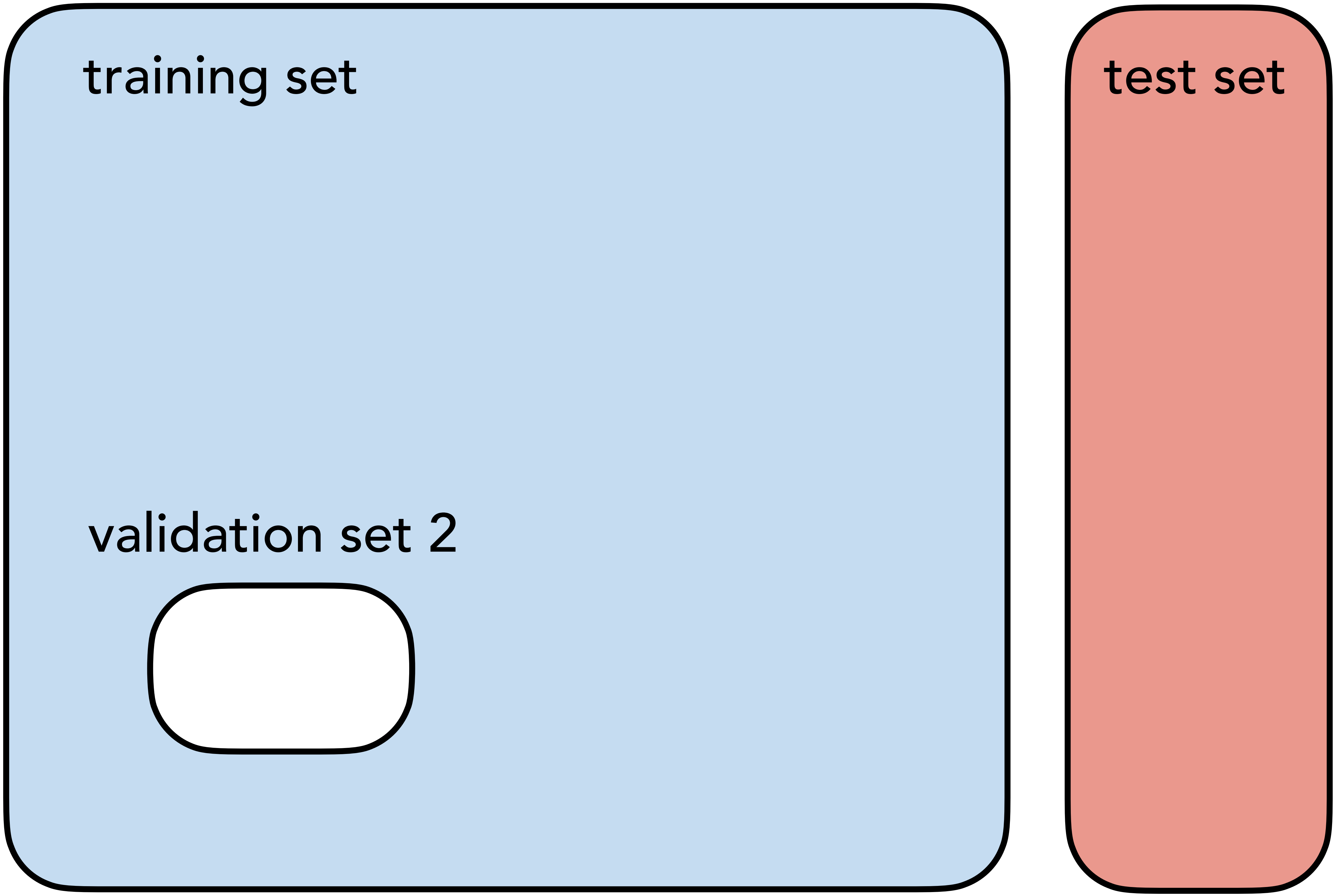
Anatomy of a Dataset



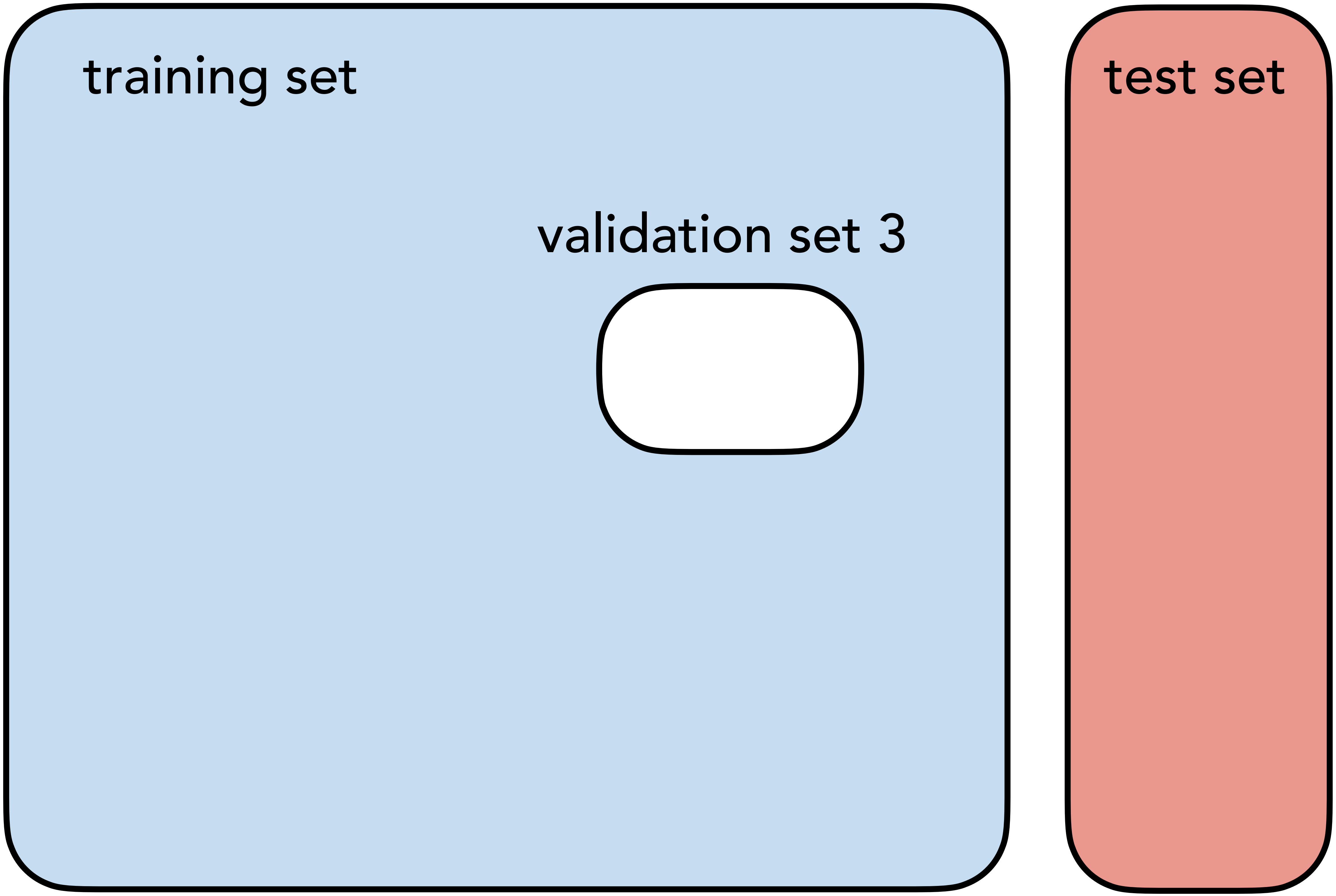
Anatomy of a Dataset



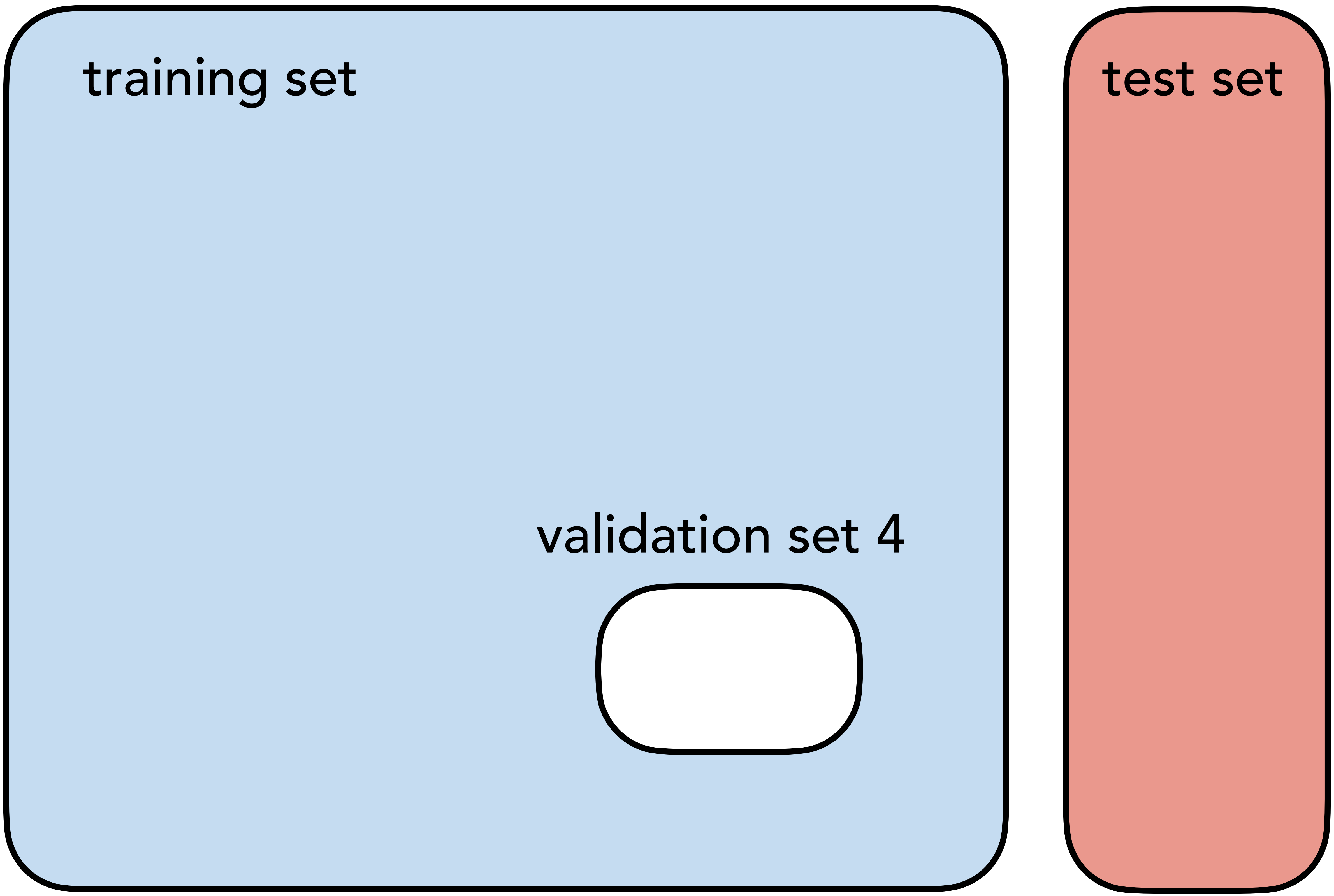
Anatomy of a Dataset



Anatomy of a Dataset



Anatomy of a Dataset



Types of learning tasks:

- Supervised learning (labeled data)
- Unsupervised learning (unlabeled data)
- Reinforcement learning (must collect data)

Supervised Learning

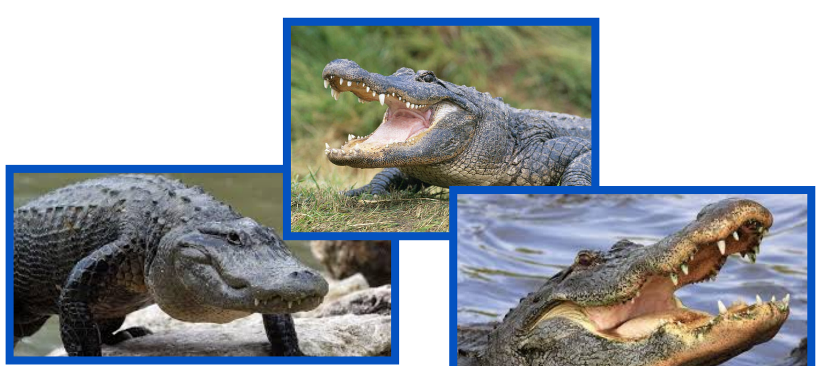
Given labeled training data (labels A and B)

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



Supervised Learning

Typical strategy:

given training set $\{\mathbf{x}_j, y_j\}$, minimize cost function

$$C = \frac{1}{N_T} \sum_j (f(\mathbf{x}_j) - y_j)^2 \qquad\qquad y_j = \begin{cases} +1 & \mathbf{x}_j \in A \\ -1 & \mathbf{x}_j \in B \end{cases}$$

by varying adjustable params of f

Cost function measures distance of trial function $f(\mathbf{x}_j)$ from idealized "indicator" function y_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
- Discover reduced representations of data for other learning tasks (e.g. supervised)

General Philosophy of Machine Learning

- Solution to problem just some function $y(\mathbf{x})$

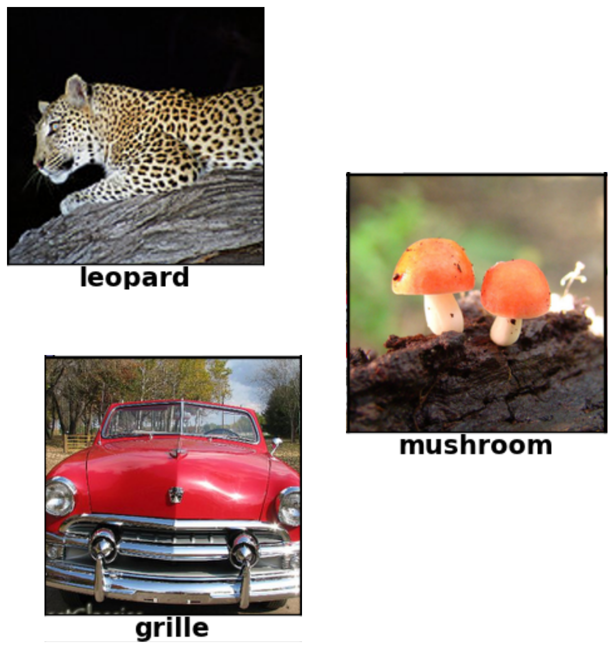
- Parameterize very flexible functions $f(\mathbf{x})$ (prefer convenient over "correct")
- Of all f that come closest to y for training data, prefer the simplest f

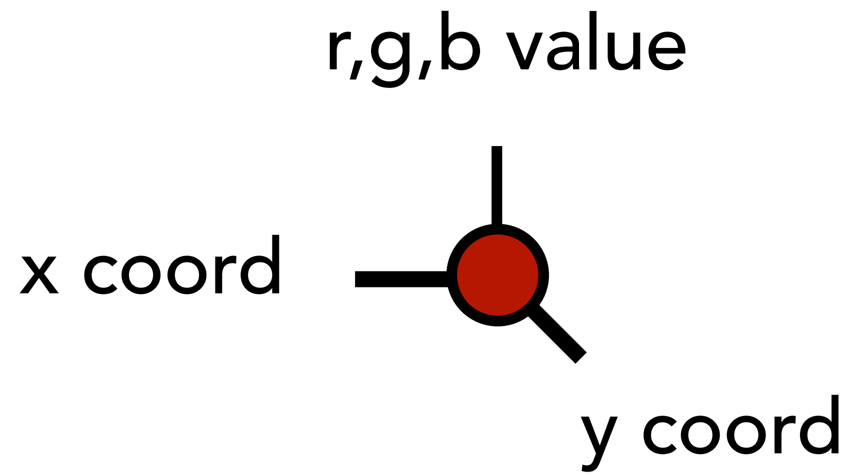
Tensors in Machine Learning

Where can tensors appear in machine learning applications?

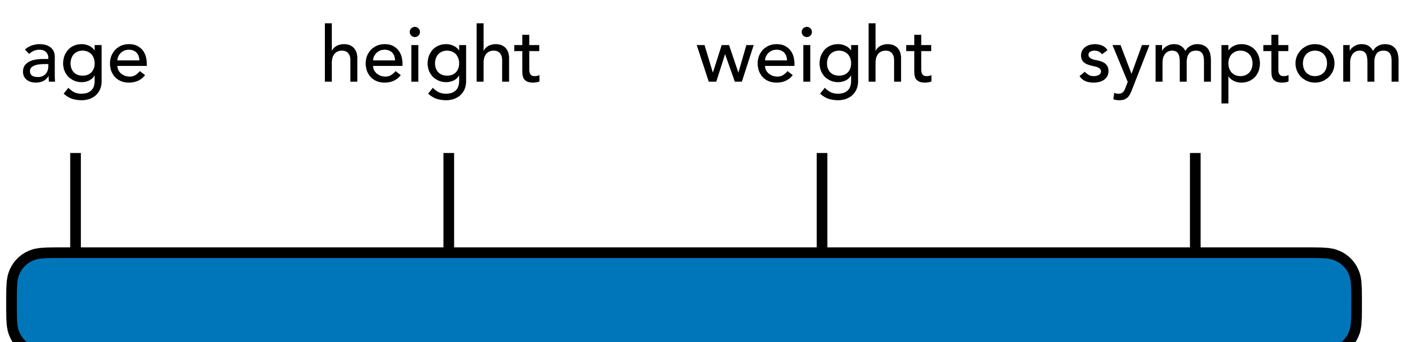
Multi-Dimensional Data

Image Data

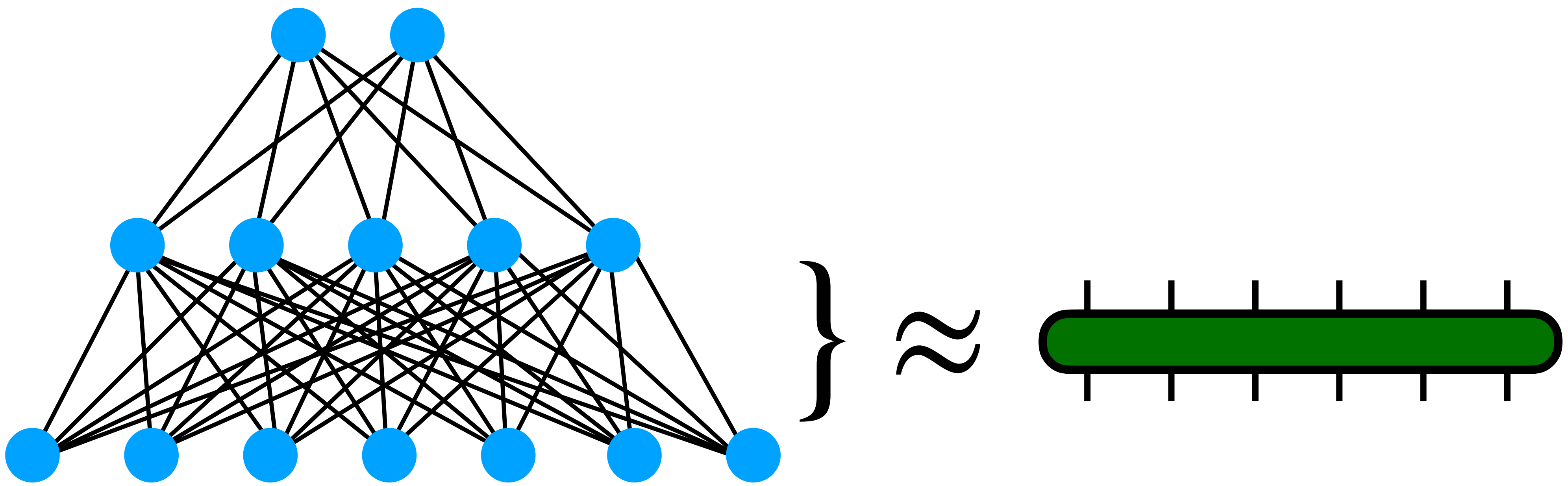




Medical Data



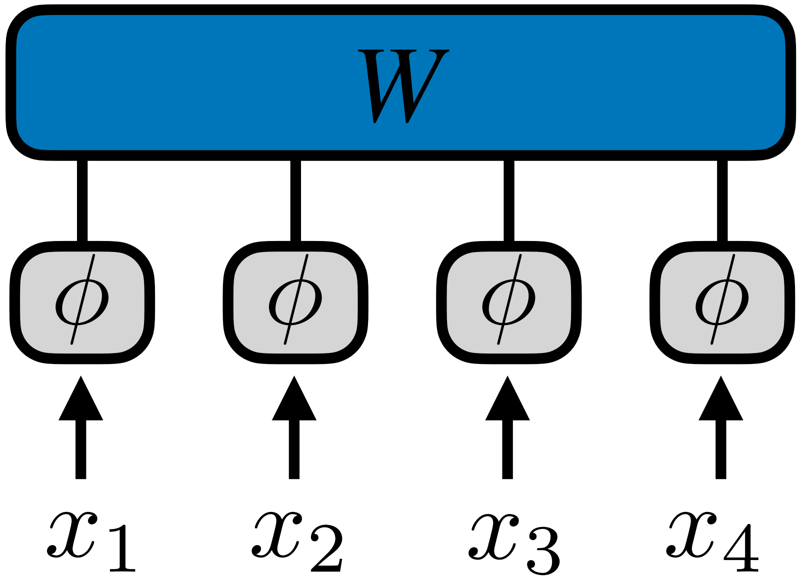
Neural Network Weight Layers



Possible to interpret as a very high-order tensor
(not just a matrix)

Weights of Discriminative, Generative Models

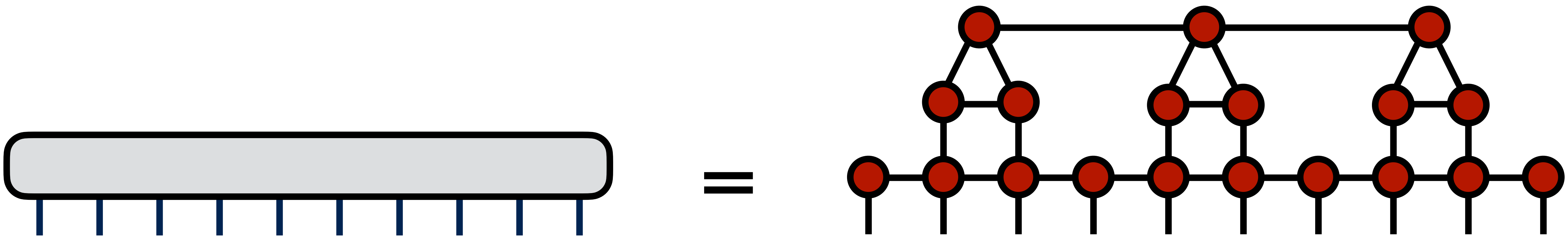
$$f(\mathbf{x}) = W \cdot \Phi(\mathbf{x})$$



For certain cases of kernel learning and Gaussian processes, weights are naturally a *high-order tensor*

Why Tensor Networks?

Tensor network = factorization of huge tensor into contracted product of smaller tensors



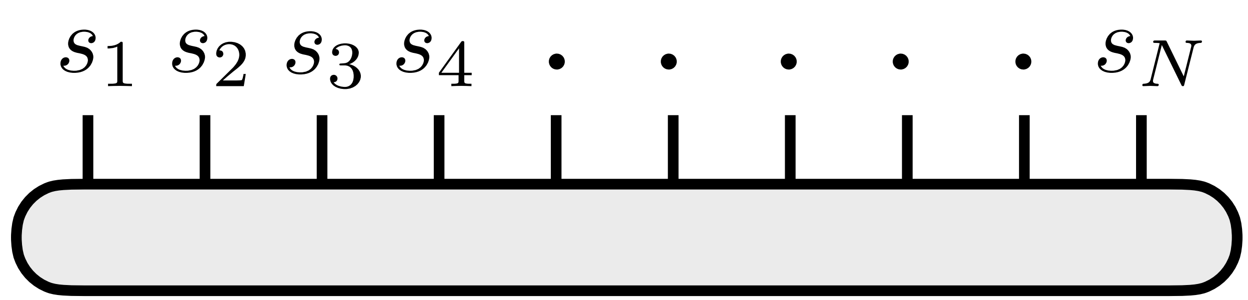
Benefits:

- exponential reduction in **memory** needed
- exponential speedup of **computations** (addition, product)
- theoretical **insight** and **interpretation**
- **estimation** of missing or corrupted entries
- many optimization **algorithms** & strategies

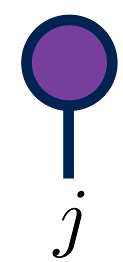
Notation – Tensor Diagrams

N-index tensor represented as shape with N lines

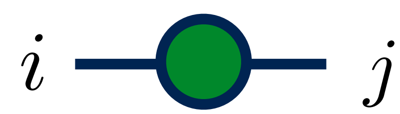
$$T^{s_1 s_2 s_3 \cdots s_N} =$$



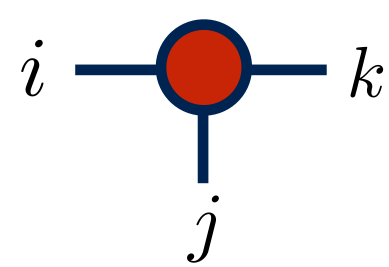
Diagrams for low-order tensors



v_j

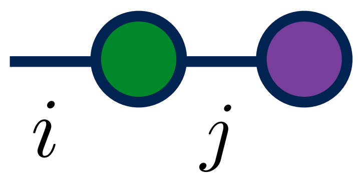


M_{ij}

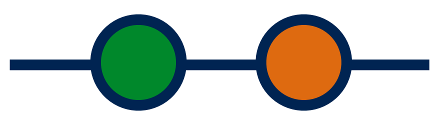


T_{ijk}

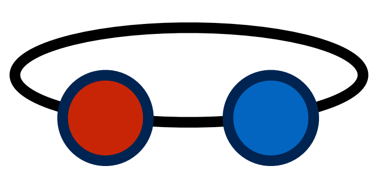
Joining lines implies contraction, can omit names



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

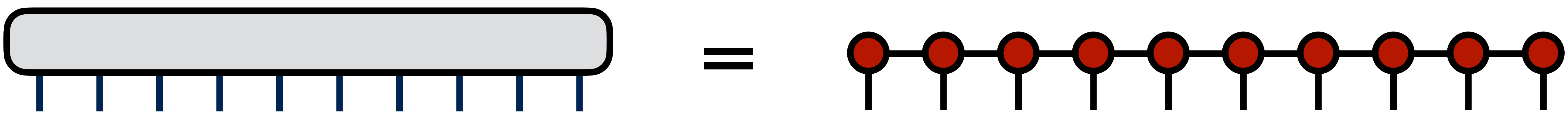


$$\longleftrightarrow \quad A_{ij} B_{jk} \quad = AB$$



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

Best understood tensor network in physics is the ***matrix product state (MPS)***[1,2]



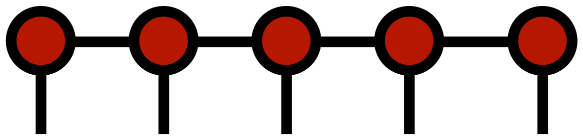
Known as ***tensor train*** in applied math literature [3]

[1] Östlund, Rommer, PRL 75, 3537 (1995)

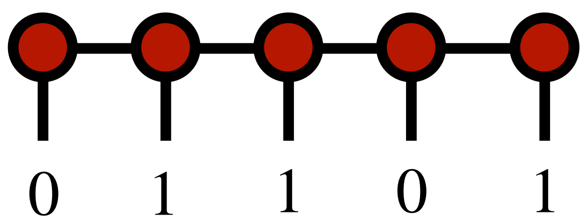
[2] Vidal, PRL 91, 147902 (2003)

[3] Oseledets, SIAM J. Sci. Comp. 33, 2295 (2011)

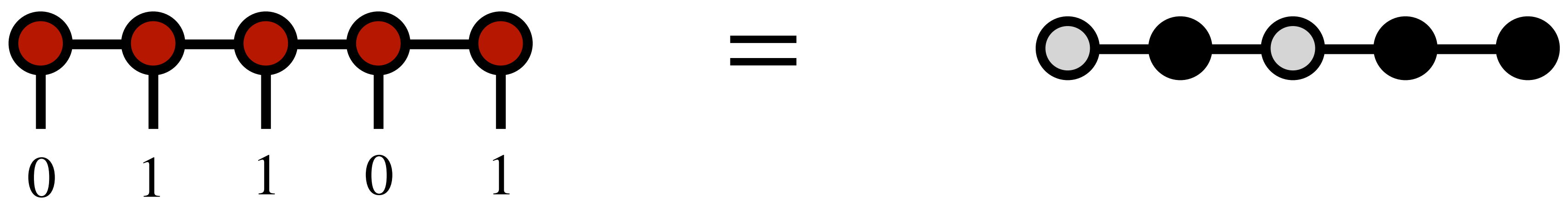
Name matrix product state refers to retrieving elements:



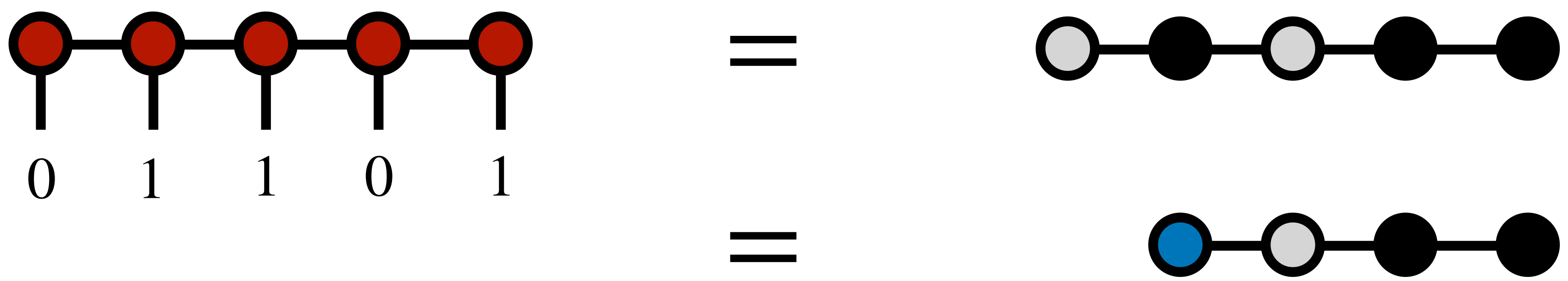
Name matrix product state refers to retrieving elements:



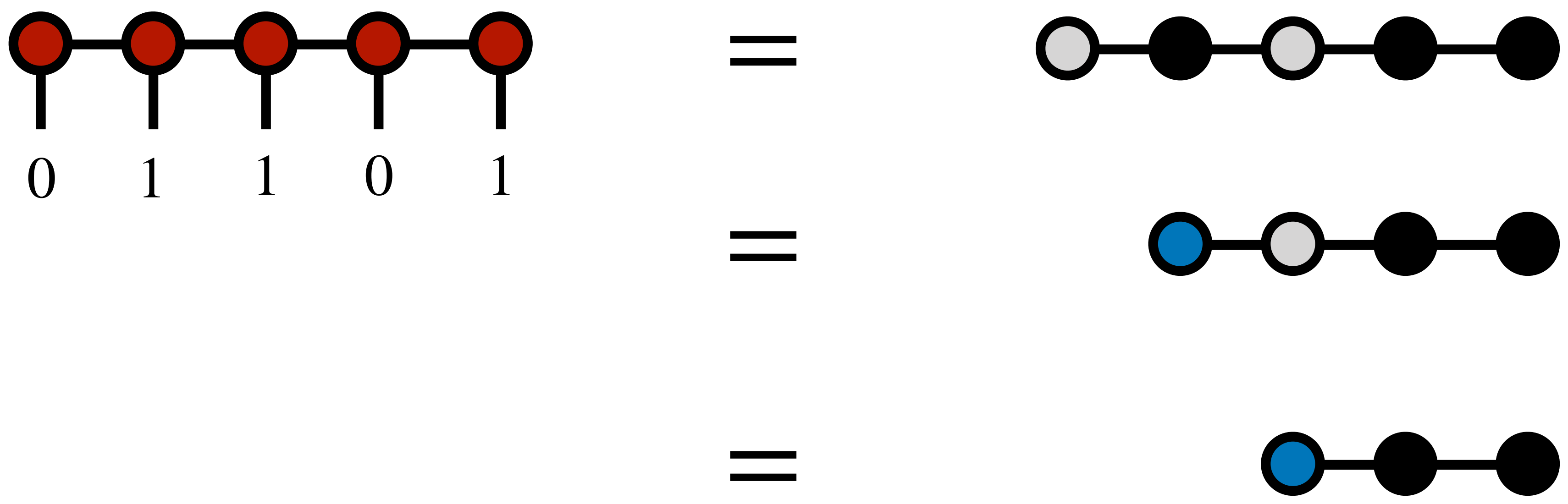
Name matrix product state refers to retrieving elements:



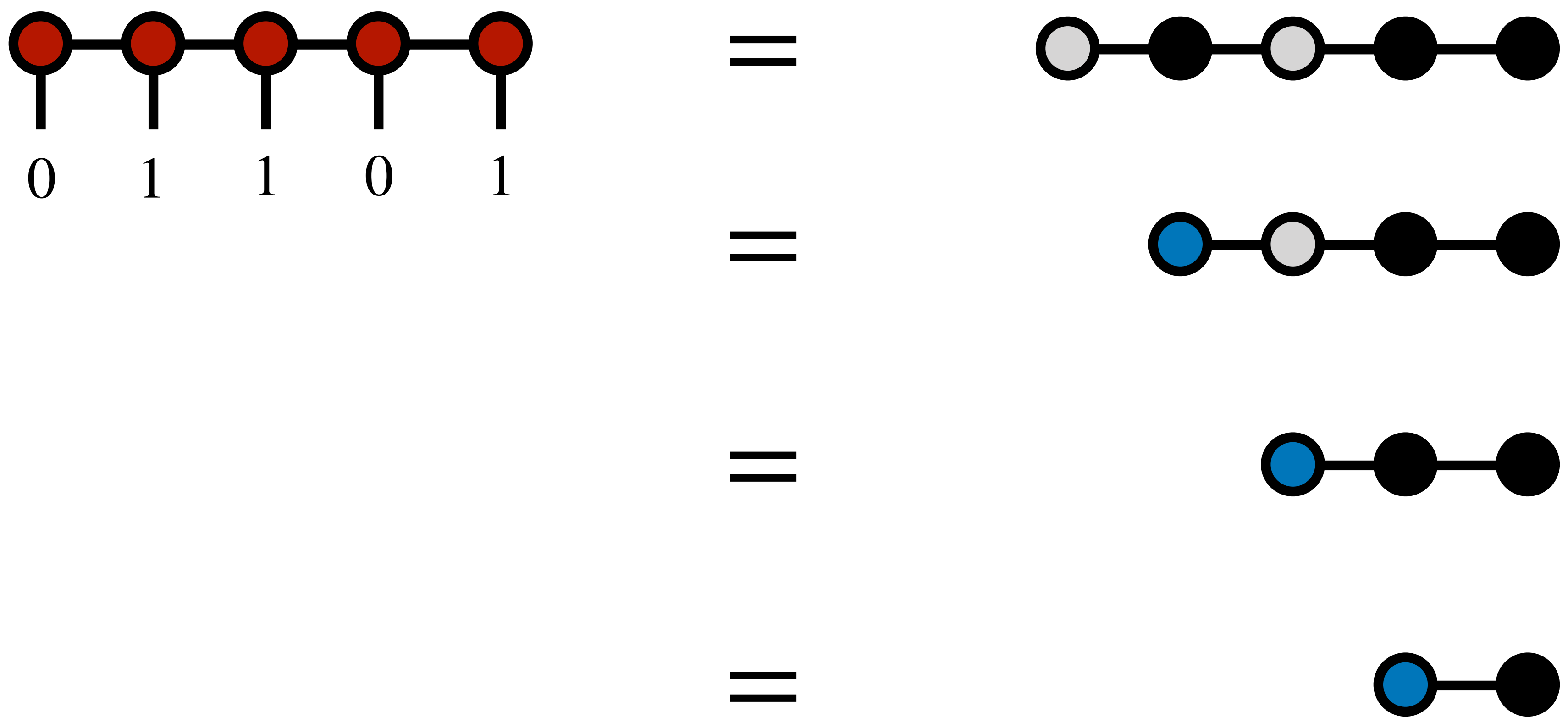
Name matrix product state refers to retrieving elements:



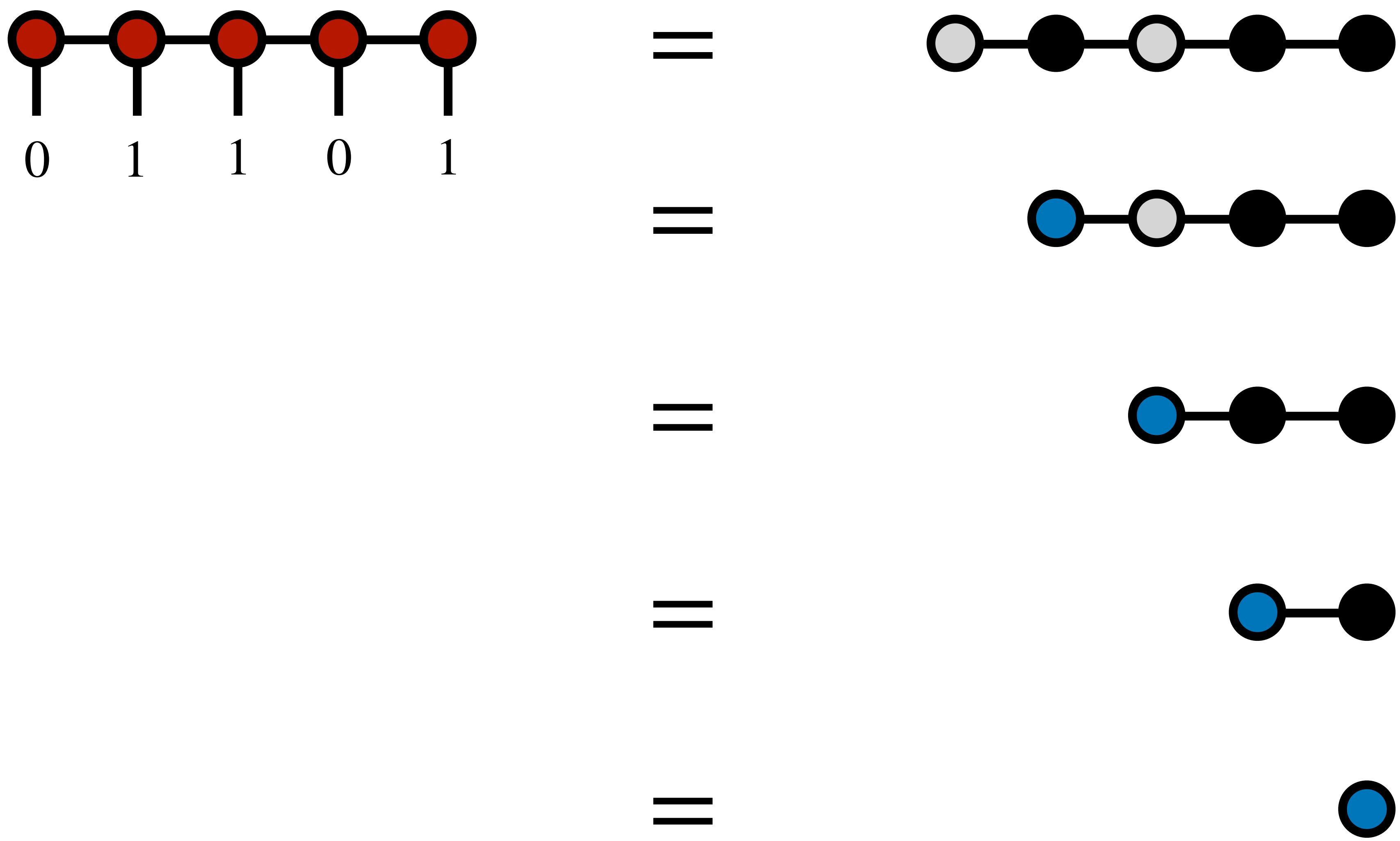
Name matrix product state refers to retrieving elements:



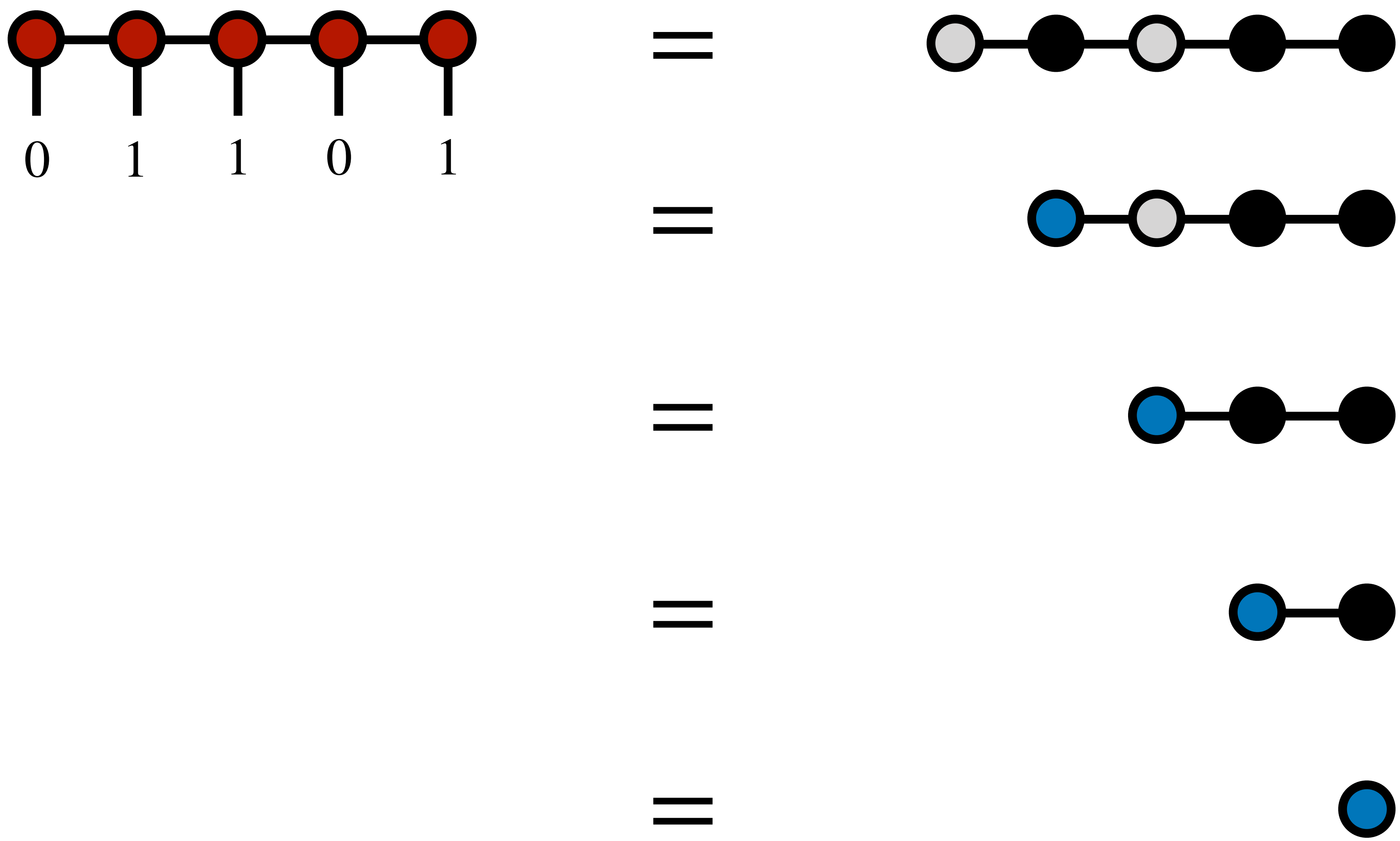
Name matrix product state refers to retrieving elements:



Name matrix product state refers to retrieving elements:

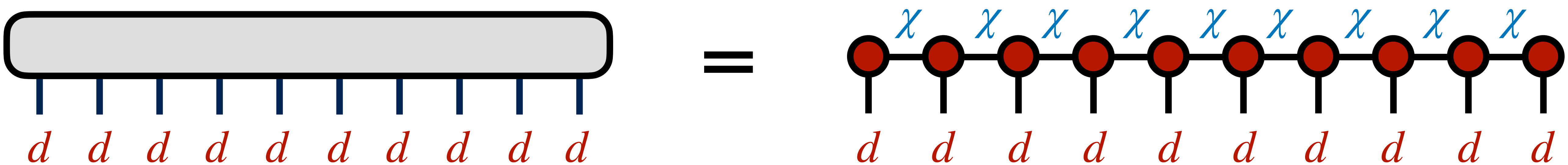


Name matrix product state refers to retrieving elements:



$$\approx \quad T^{01101}$$

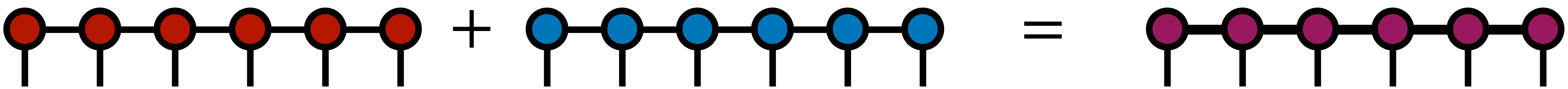
Adjustable parameter of matrix product state (MPS) is bond dimension χ



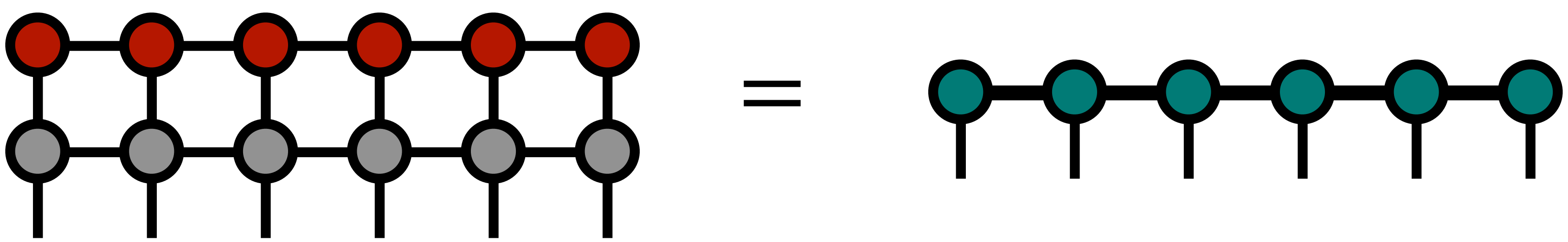
If modest χ yields good approximation, obtain massive compression:

$$d^N \longrightarrow N\, d\, \chi^2$$

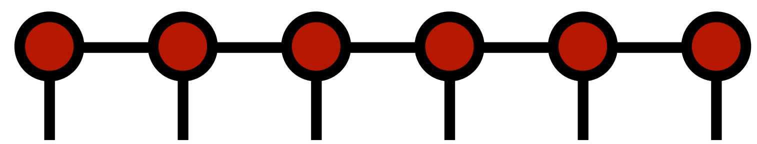
Can efficiently sum MPS compressed form:



Or multiply by other networks:

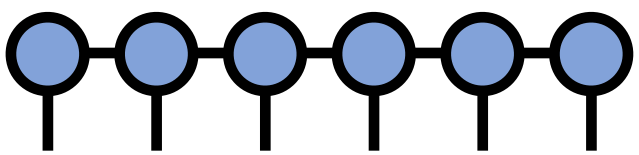


MPS can be perfectly sampled (no autocorrelation):

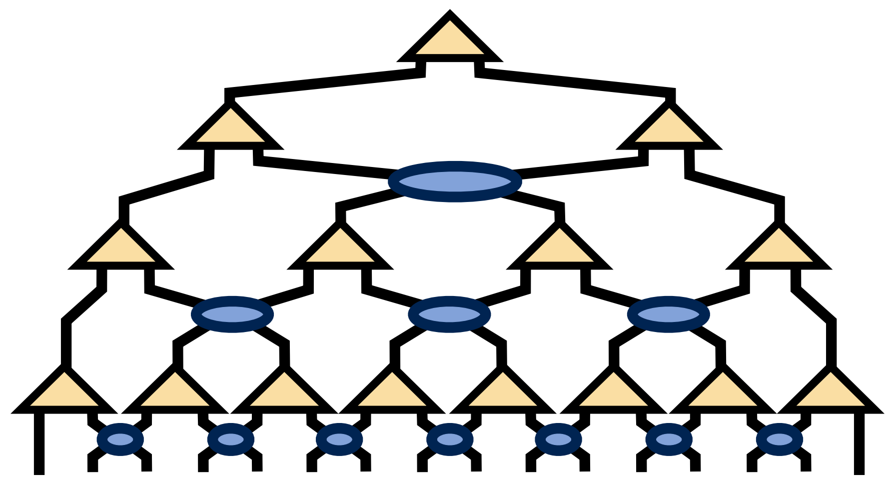


0	0	0	0	0	0
0	1	0	0	1	0
0	1	1	0	0	0
0	0	1	0	0	1
1	1	0	1	0	1
0	0	0	0	0	0
0	1	1	0	0	0

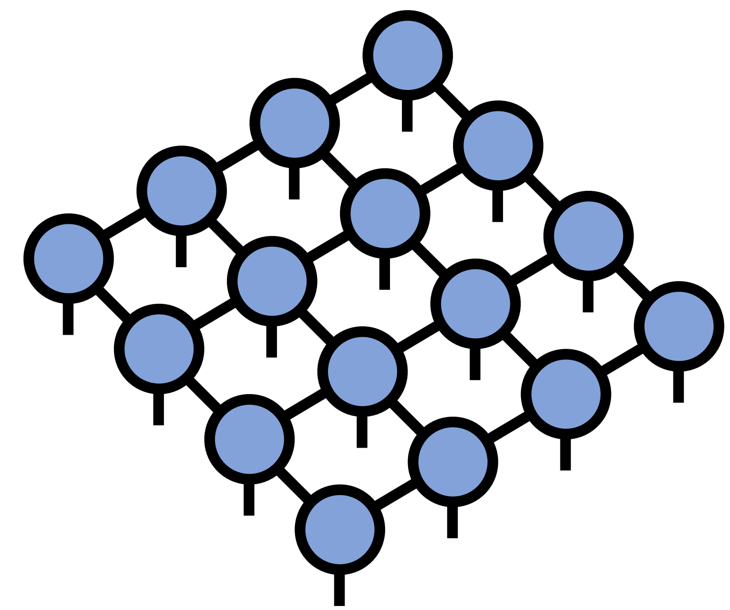
In quantum physics context, have rich theory of which tensor networks are suited for particular "data"



1D system



1D *critical* system

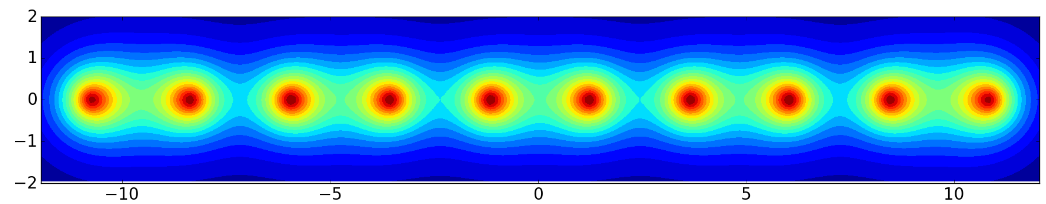


2D system

(Here "data" = samples/measurements of a quantum wavefunction)

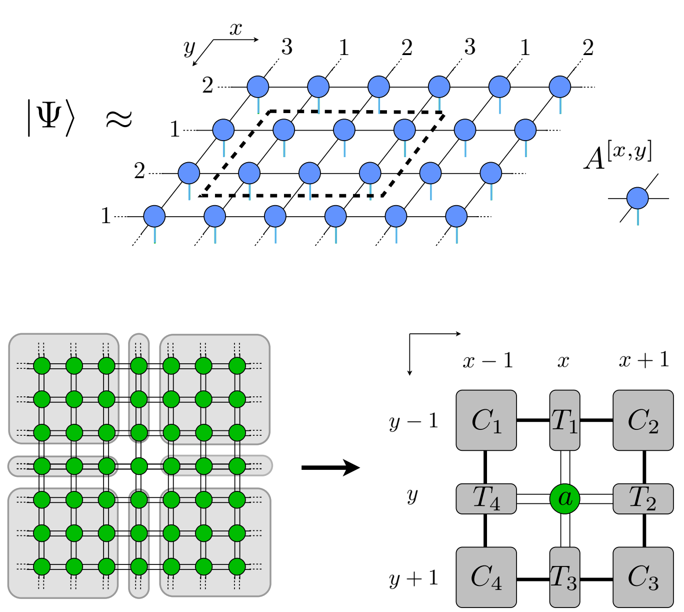
Last but not least, can pull off impressive calculations

capture wavefunctions of 1000's of atoms

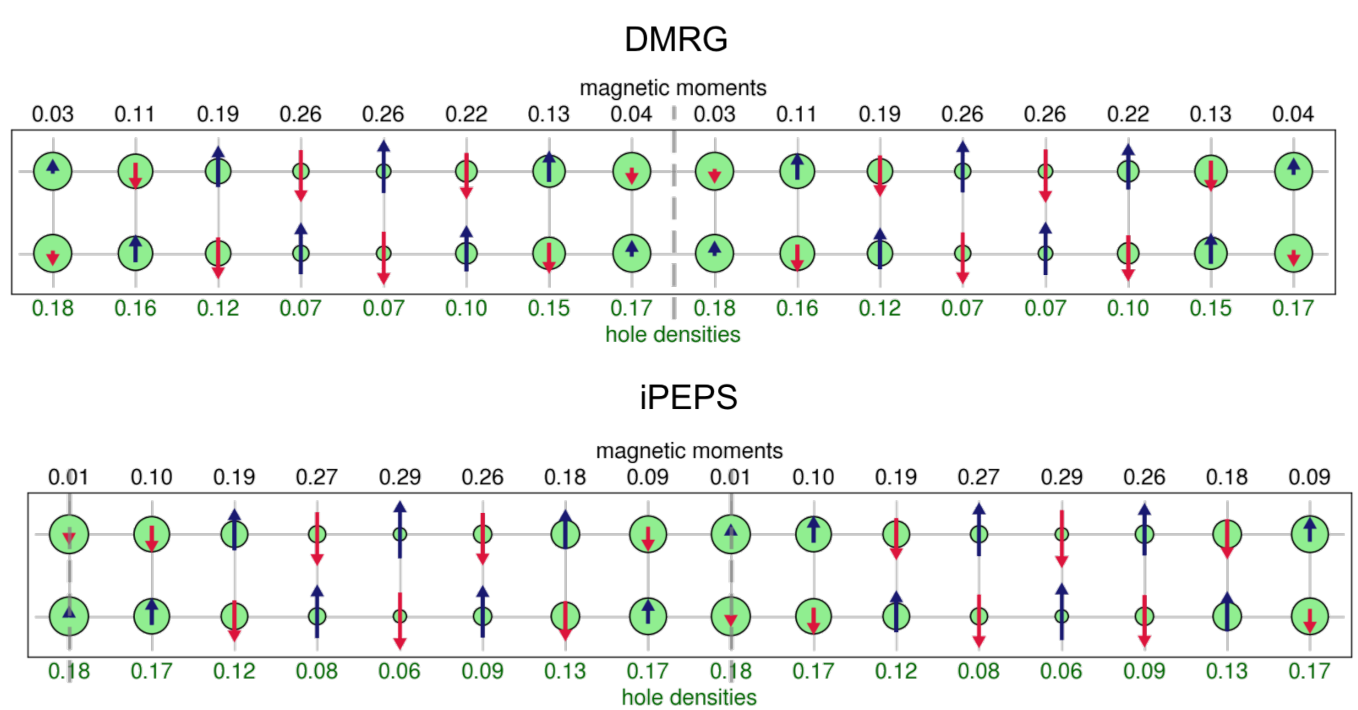


Stoudenmire, White, PRL 119, 046401 (2017)

study challenging, correlated electron models in 2D



Corboz, PRB 94, 035133 (2016)



Zheng et al., Science 358, 1155 (2017)

Are tensor networks only useful for wavefunctions?
Why not other tensors too?

Actually, since late 2000's, applied math community exploring tensor networks (Hackbusch, Oseledets,...)

Comparative study of Discrete Wavelet Transforms and Wavelet Tensor Train decomposition to feature extraction of FTIR data of medicinal plants

Pavel Kharyuk[a,b], Dmitry Nazarenko[c,*], Ivan Oseledets[a,d]

Principal Component Analysis with Tensor Train Subspace

Wenqi Wang, Vaneet Aggarwal, and Shuchin Aeron

A Tensor-Train Deep Computation Model for Industry Informatics Big Data Feature Learning

4 Author(s) Qingchen Zhang ; Laurence T. Yang ; Zhikui Chen ; Peng Li **View All Authors**

Tensor networks a general tool for linear algebra in exponentially high-dimensional spaces

Notions like entanglement entropy correspond to multilinear tensor rank

MPS and other tensor networks already being explored for machine learning:

supervised learning

Novikov, Trofimov, Oseledets, arxiv:1605.03795

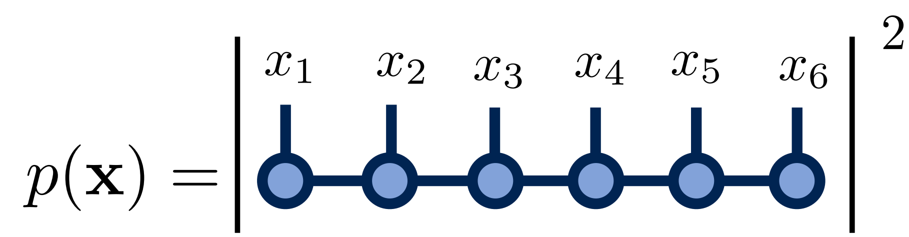
Stoudenmire, Schwab, *Advances in N.I.P.S.*, **29**, 4799

I. Glasser, N. Pancotti, J.I. Cirac, arxiv:1806.05964

Efthymiou, et al., arxiv:1906.06329

Selvan, et al., arxiv:2004.10076

unsupervised learning



Bengua, Phien, Tuan, 10.1109/BigDataCongress.2015.105 (2015)

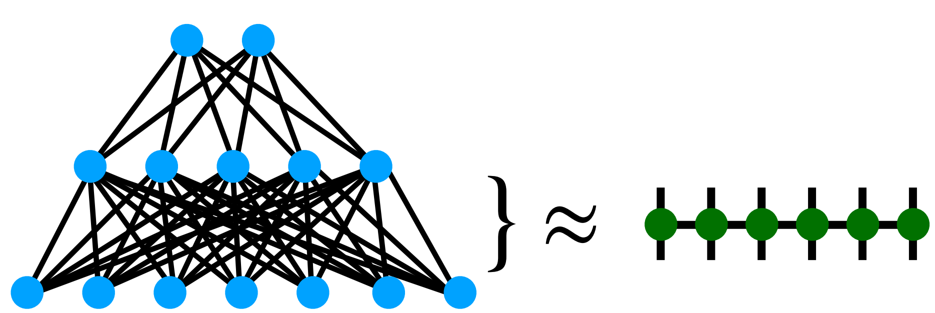
Cichocki et al, dx.doi.org/10.1561/2200000067 (2017)

Han, et al, Phys. Rev. X 8, 031012 (2018)

Stokes, Terilla, arxiv:1902.06888

Miller, et al., arxiv:2003.01039

compressing neural nets



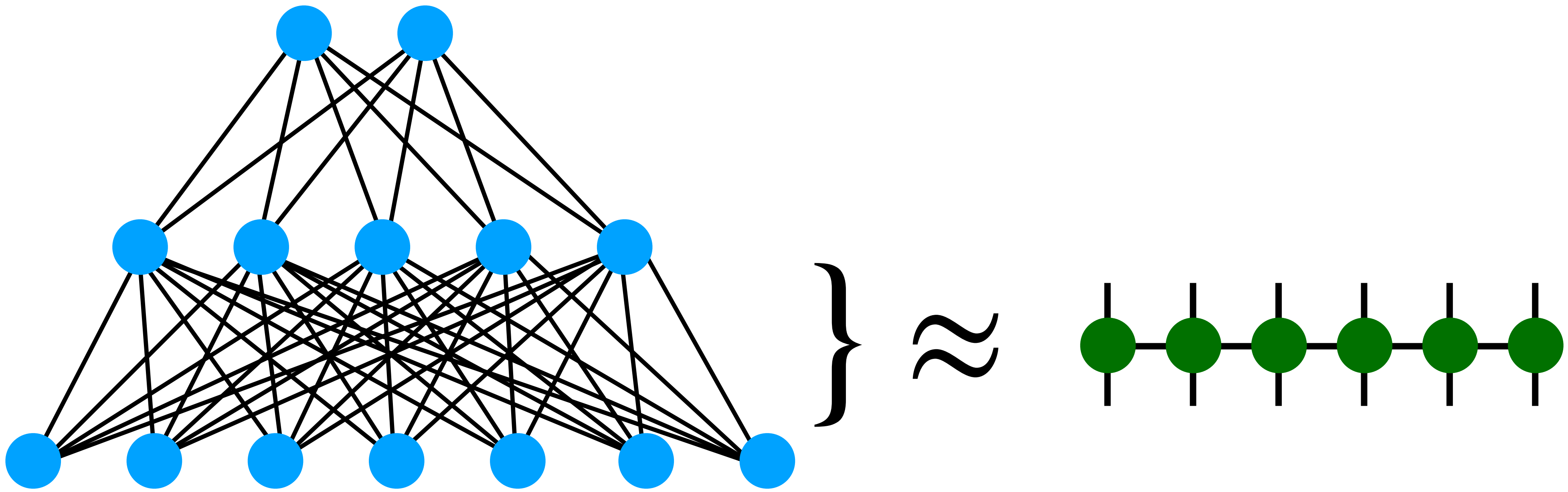
Novikov et al., Advances in NIPS (2015), arxiv:1509.06569

Garipov, Podoprikhin, Novikov, arxiv:1611.03214

Yu, Zheng, Anandkumar, , arxiv:1711.00073

...and many others

Compressing Neural Network Weight Layers



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

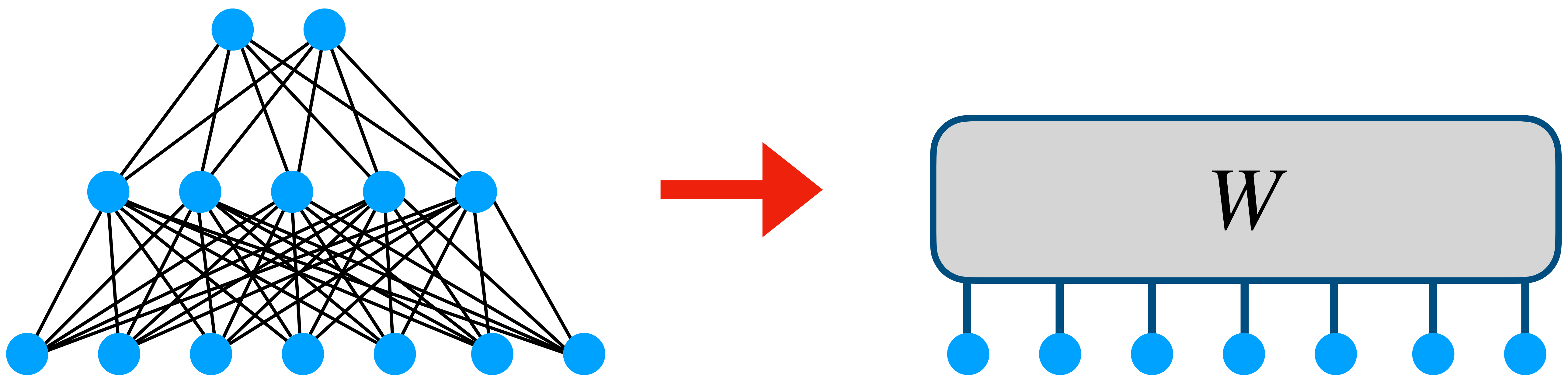
Garipov, Podoprikhin, Novikov, arxiv:1611.03214

Hallam, Andrew, et al. "Compact neural networks based on the multiscale entanglement renormalization ansatz." arXiv:1711.03357

Rose Yu, Stephan Zheng, Anima Anandkumar, Yisong Yue, "Long-term Forecasting using Tensor-Train RNNs", arXiv:1711.00073

- Train very "wide" model: 262,144 hidden units
- Achieve 80x compression, only 1% accuracy loss

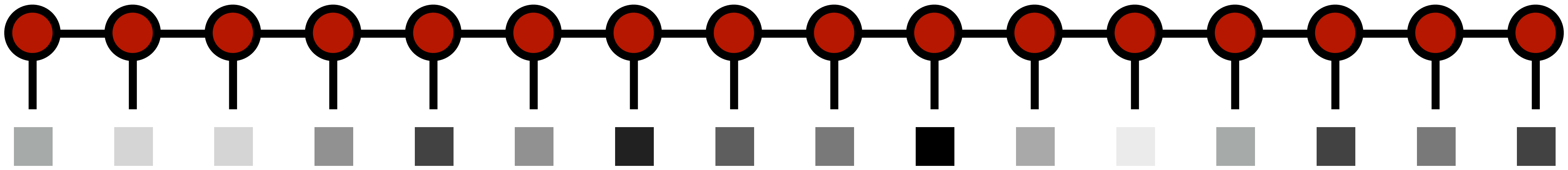
Framework where tensor network plays central role?



Motivation:

- Can natural images be more complex than wavefunctions?
- Import many ideas, algorithms from physics
- Improve tensor network methods

MPS and Tensor Networks for General Data



Ingredients for machine learning:

Data (+ labels): $\{\mathbf{x}_j, y_j\}$

Objective / cost function: $$\min_f \frac{1}{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{T}|} \Big(f(\mathbf{x}_j) - y_j \Big)^2$$

Class of model functions: $f(\mathbf{x})$

How to get a class of functions where a huge (order-N) tensor appears?

Consider a polynomial over N variables:

$$f(x_1, x_2, \ldots, x_N) = \sum_{\mathbf{n}} W_{n_1 n_2 \cdots n_N} \, x_1^{n_1} x_2^{n_2} \cdots x_N^{n_N}$$

Novikov, Trofimov, Oseledets, arxiv:1605.03795
Stoudenmire, Schwab, arxiv:1605.05775

How to get a class of functions where a huge (order-N) tensor appears?

Consider a polynomial over N variables:

$$f(x_1, x_2, \ldots, x_N) = \sum_{\mathbf{n}} W_{n_1 n_2 \cdots n_N} \, x_1^{n_1} x_2^{n_2} \cdots x_N^{n_N}$$

$$= W_{00000} + W_{10000} \; x_1 + W_{01000} \; x_2 + \ldots$$

$$+ W_{11000} \; x_1 x_2 + W_{10100} \; x_1 x_3 + \ldots$$

$$+ W_{11111} \; x_1 x_2 x_3 x_4 x_5$$

Novikov, Trofimov, Oseledets, arxiv:1605.03795
Stoudenmire, Schwab, arxiv:1605.05775

More generally, use any basis of products of functions

$$f(x_1, x_2, \ldots, x_N) = \sum_{\mathbf{n}} W_{n_1 n_2 \cdots n_N} \; \phi^{n_1}(x_1)\phi^{n_2}(x_2)\cdots\phi^{n_N}(x_N)$$

Novikov, Trofimov, Oseledets, arxiv:1605.03795
Stoudenmire, Schwab, arxiv:1605.05775

More generally, use any basis of products of functions

$$f(x_1, x_2, \ldots, x_N) = \sum_{\mathbf{n}} W_{n_1 n_2 \cdots n_N} \; \phi^{n_1}(x_1)\phi^{n_2}(x_2)\cdots\phi^{n_N}(x_N)$$

$$\phi(x) = \begin{bmatrix} 1 \\ x \end{bmatrix}$$ gives polynomials

Novikov, Trofimov, Oseledets, arxiv:1605.03795
Stoudenmire, Schwab, arxiv:1605.05775

More generally, use any basis of products of functions

$$f(x_1, x_2, \ldots, x_N) = \sum_{\mathbf{n}} W_{n_1 n_2 \cdots n_N} \; \phi^{n_1}(x_1)\phi^{n_2}(x_2)\cdots\phi^{n_N}(x_N)$$

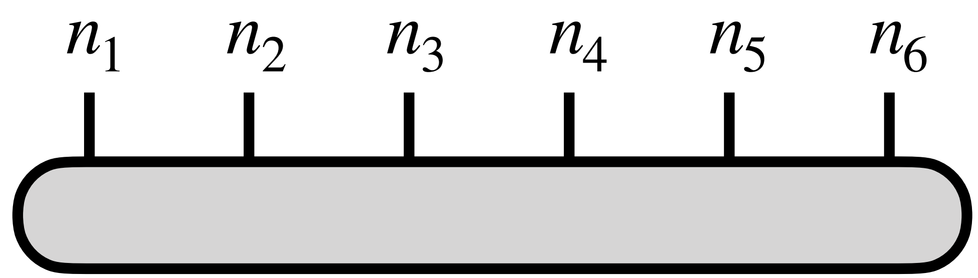
$$\phi(x) = \begin{bmatrix} 1 \\ x \end{bmatrix}$$ gives polynomials

$$\phi(x) = \begin{bmatrix} \cos(\pi x/2) \\ \sin(\pi x/2) \end{bmatrix}$$ gives symmetry $x \leftrightarrow (1-x)$

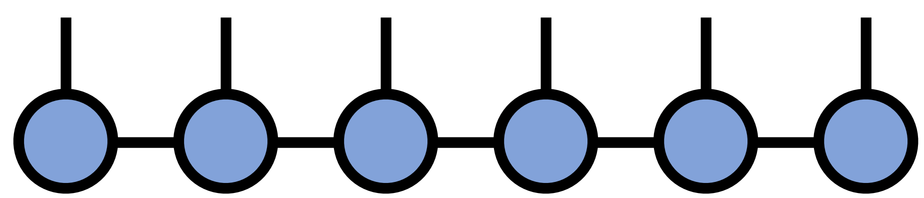
Novikov, Trofimov, Oseledets, arxiv:1605.03795
Stoudenmire, Schwab, arxiv:1605.05775

Main idea: factorize weight tensor

$$W_{n_1 n_2 n_3 n_4 n_5 n_6} =$$

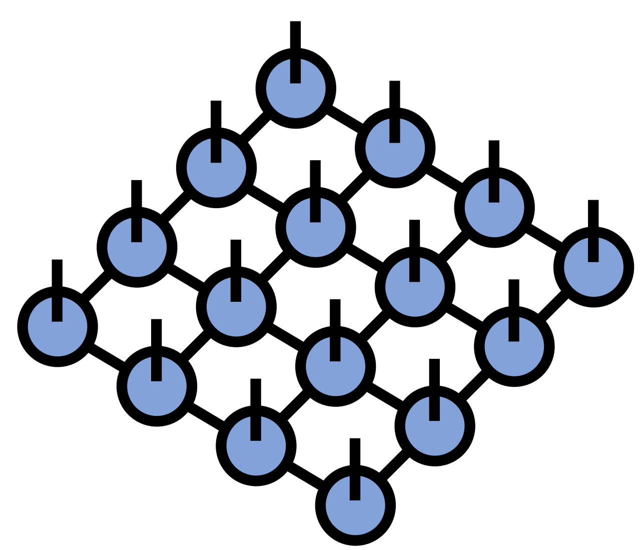


$$=$$



(or

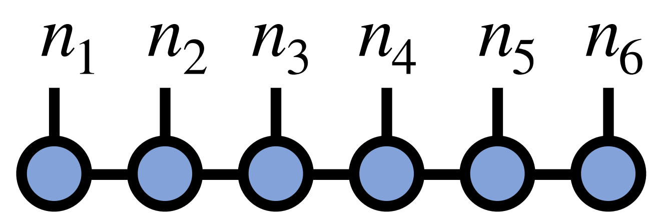
$$=$$



) or other tensor networks

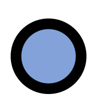
Can use as a model class for supervised or unsupervised learning

$$f(x_1, x_2, \ldots, x_6) = \sum_{\mathbf{n}}$$



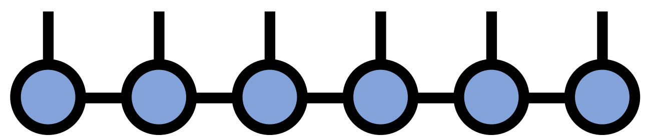
$$x_1^{n_1} x_2^{n_2} \cdots x_6^{n_6}$$

$$\min \frac{1}{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{T}|} \Big(f(\mathbf{x}_j) - y_j \Big)^2$$



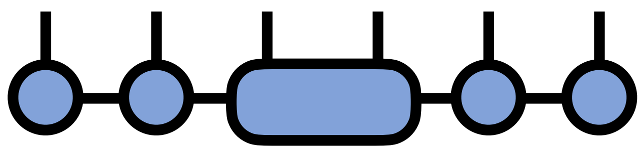
Immediate payoff: adaptivity of weights

1) merge (contract) a pair of tensors



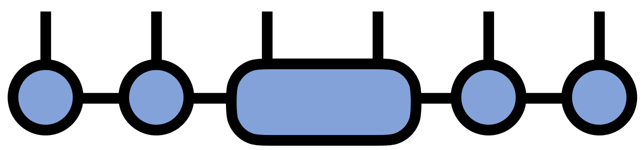
Immediate payoff: adaptivity of weights

1) merge (contract) a pair of tensors

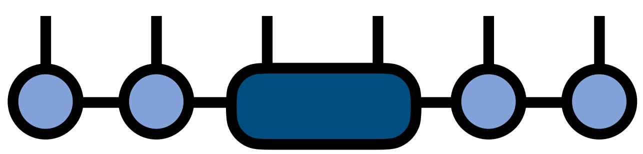


Immediate payoff: adaptivity of weights

1) merge (contract) a pair of tensors

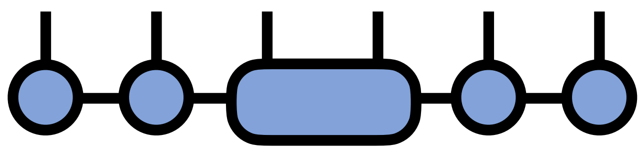


2) optimize parameters (e.g. gradient steps)

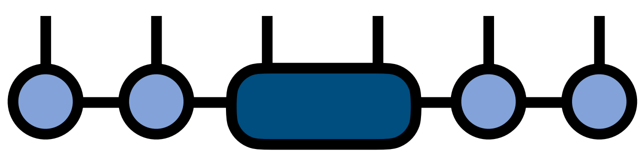


Immediate payoff: adaptivity of weights

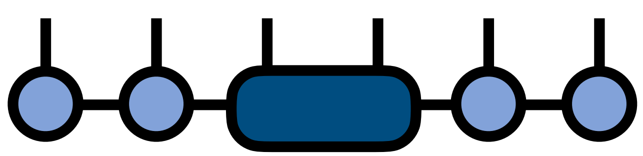
1) merge (contract) a pair of tensors



2) optimize parameters (e.g. gradient steps)

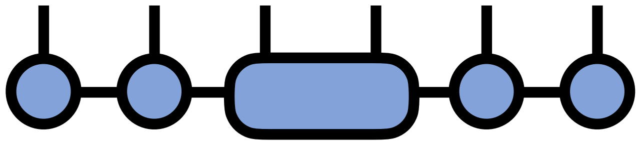


3) SVD factorization to adapt size of bond index

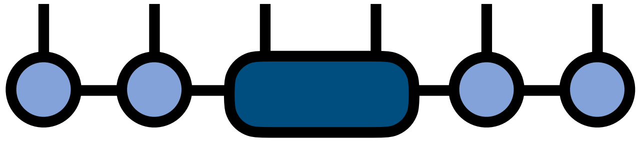


Immediate payoff: adaptivity of weights

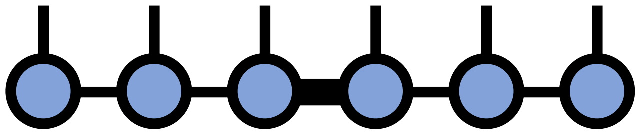
1) merge (contract) a pair of tensors



2) optimize parameters (e.g. gradient steps)



3) SVD factorization to adapt size of bond index



How well do tensor networks perform for supervised & unsupervised learning?

What applications can be done?

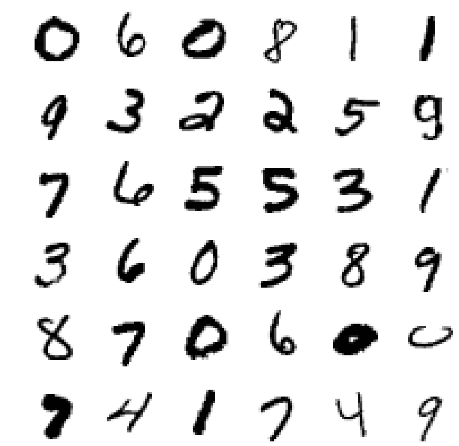
Supervised Learning

MPS have been used as proof-of-principle for tensor networks in supervised learning

1. MNIST data set (99% test accuracy)

   Stoudenmire, Schwab, NIPS 29, 4799 (2016)

   Efthymiou, Hidary, Leichenauer, arxiv:1906.06329



2. Fashion MNIST data set (89% test accuracy)

   Stoudenmire, Quant. Sci. Tech. 3, 034003 (2018)

   Glasser, Pancotti, Cirac, arxiv:1806.05964 (2018)

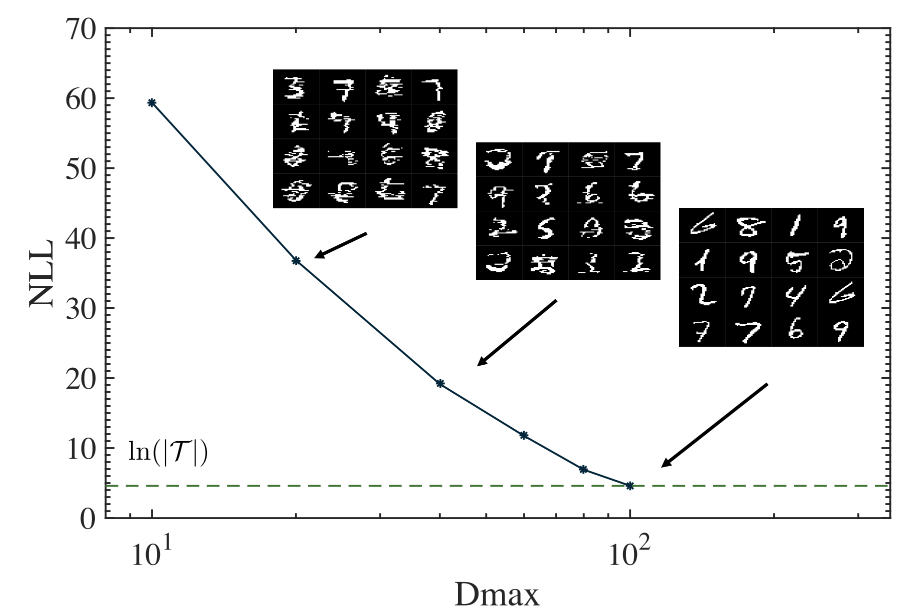
Generative Modeling

MPS have been used to parameterize generative models

1. Trained by gradient optimization of log-likelihood

MNIST data set

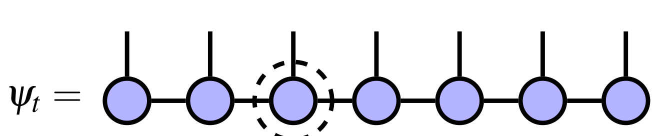
*Han, Wang, Fan, Wang, Zhang, PRX **8**, 031012 (2018)*
*Cheng, Wang, Xiang, Zhang, PRB **99**, 155131 (2019)*

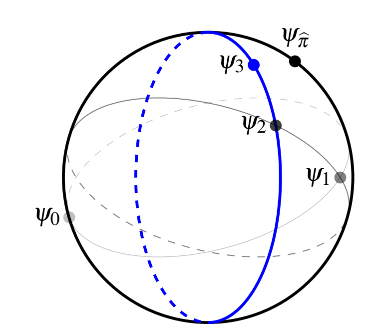


2. Trained by local-exact-solution algorithm

parity (N=20) data set

Stokes, Terilla, arxiv:1902.06888





Concatenating multiple MPS (string-bond) and pre-processing with a CNN layer gives state-of-the-art on Fashion MNIST:

Method	Test Accuracy
AlexNet	88.90%
Tree tensor net. [1]	88.97%
String bond state [2]	89.2%
CNN-String bond state [2]	92.3%
GoogLeNet	93.7%

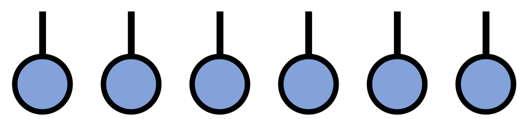
fashion MNIST

Glasser, Pancotti, Cirac, arxiv:1806.05964

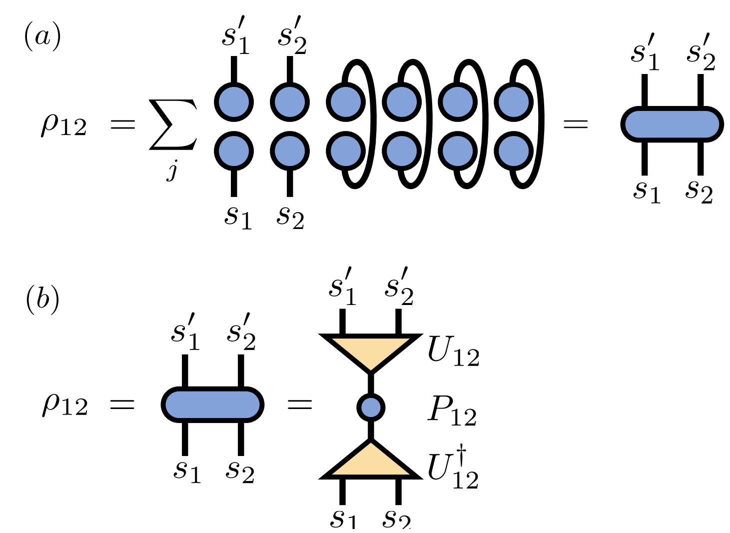
Extension #1:
hierarchical optimization of tensor network
unsupervised / supervised hybrid

Data feature map:

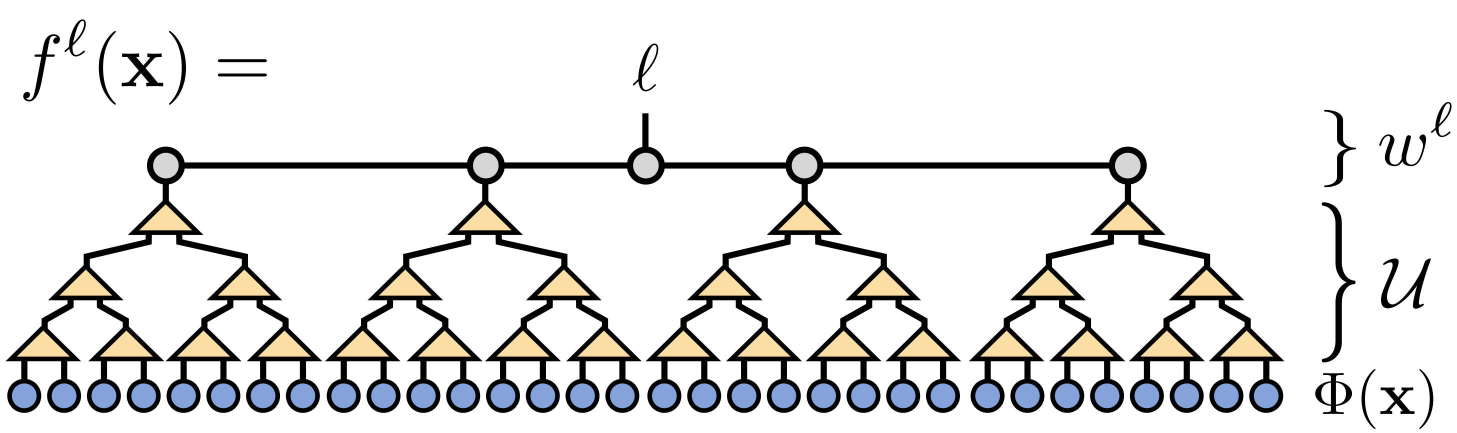
$$\mathbf{x} \longrightarrow \Phi(\mathbf{x}) =$$



Deterministic tensor learning:



Resulting model – train top tensors for supervised objective:

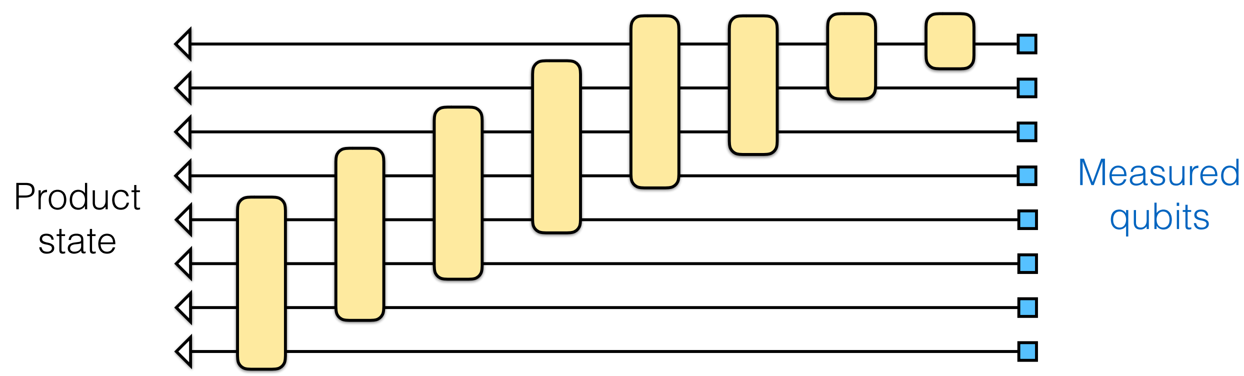


89% accuracy on Fashion MNIST data set

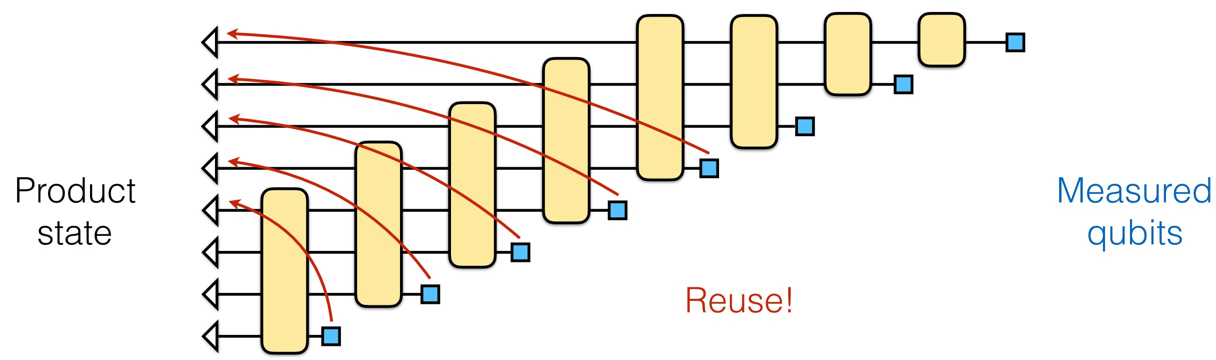
Stoudenmire, Quant. Sci. Tech. 3, 034003 (2018) [arxiv:1801.00315]

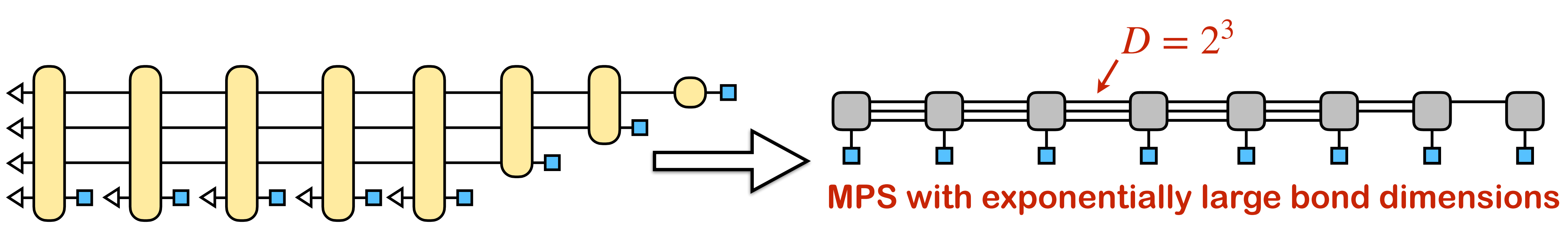
Extension #2:

algorithm for quantum machine learning
identical to optimization of a tensor network!



qubit efficient scheme:





Huggins, Patel, Whaley, Stoudenmire, *Quant. Sci. Tech.* **4**, 024001 (2019) [arxiv:1803.11537]

Related idea: <u>infinite</u> DMRG on quantum computers with finite number of physical qubits

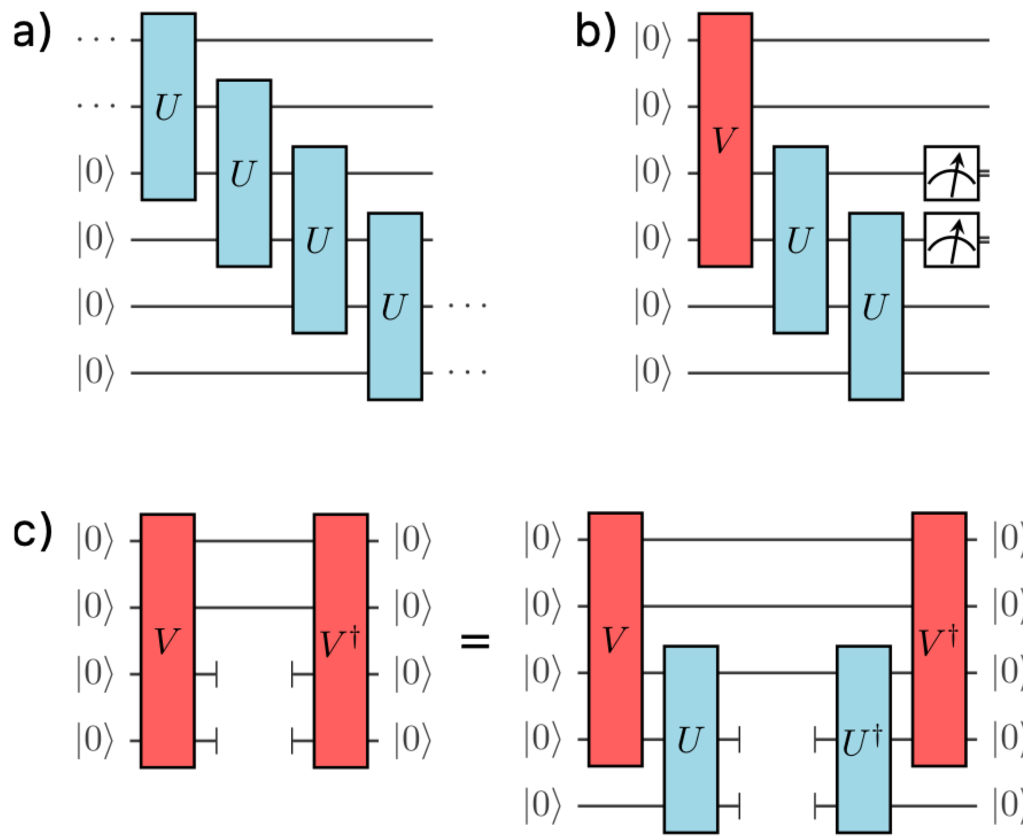


FIG. 2. **Quantum circuits for translationally invariant states and their local measurement.** a) An infinite

Some of these techniques could be used for machine learning too

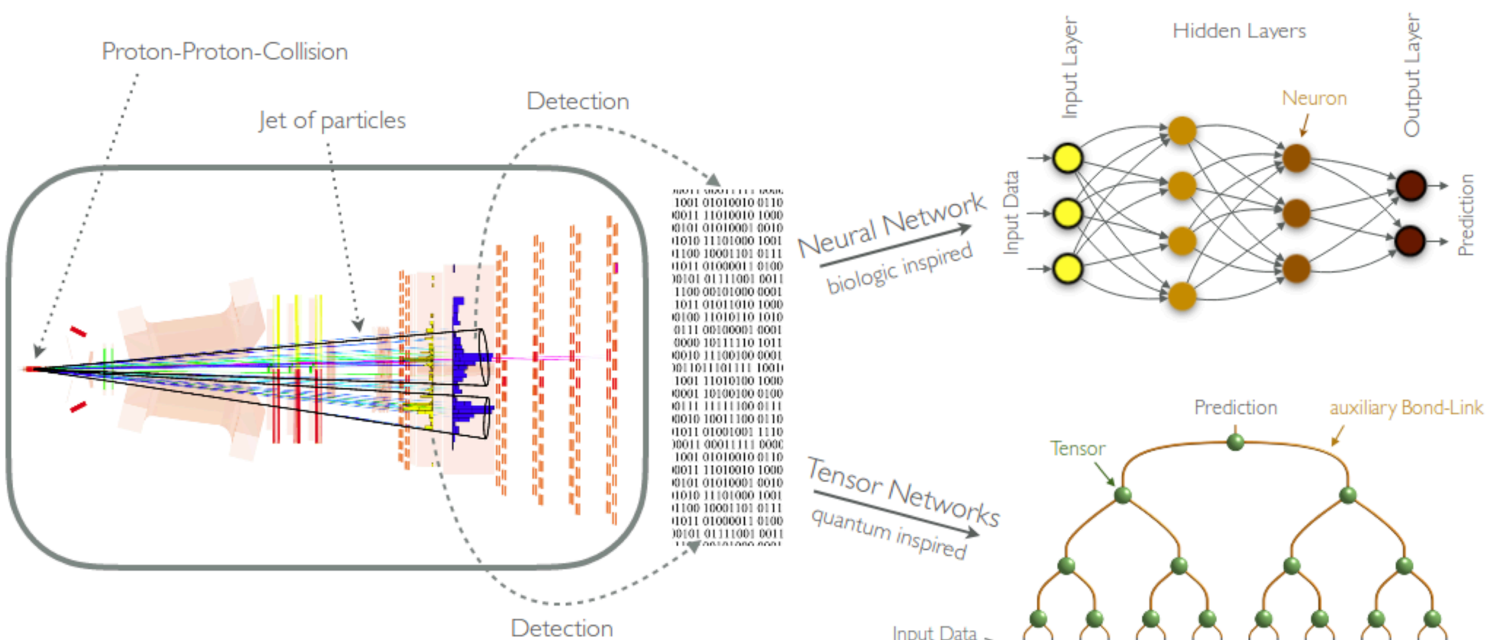
Barratt, Dborin, Bal, Stojevic, Pollmann, Green, arxiv: 2003.12087

Review: Notable Recent Works Using *Tensor Network Machine Learning*

Quantum-Inspired Machine Learning on High-Energy Physics Data

Marco Trenti, Lorenzo Sestini, Alessio Gianelle, Davide Zuliani, Timo Felser, Donatella Lucchesi, Simone Montangero, arxiv:2004.13747

- Used tree tensor network
- Determine charge of quark from events
- Compared to neural networks and gave excellent results

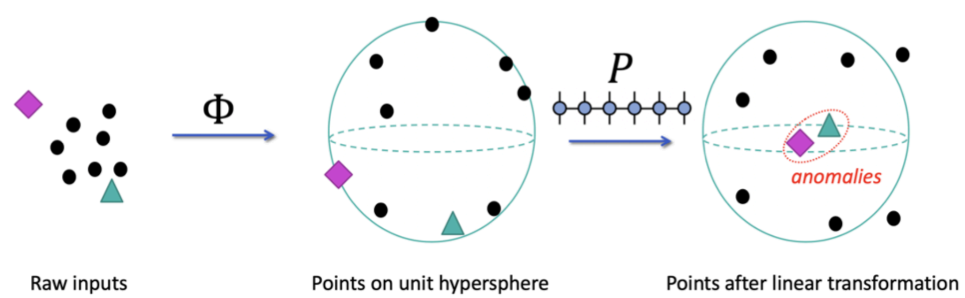


Schematic Learning Process

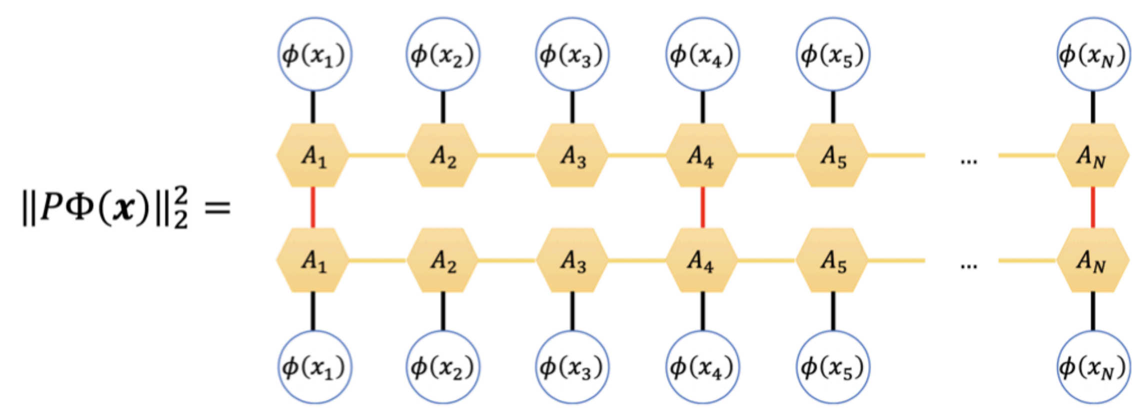
Anomaly Detection with Tensor Networks

Jinhui Wang, Chase Roberts, Guifre Vidal, Stefan Leichenauer, arxiv:2006.02516

- Used squared tensor network (locally purified state)
- Perform anomaly detection task
- Strongly competes with, and for tabular data **exceeds performance** of state-of-art neural net approaches



Geometric Anomaly Detection



Model Architecture

Table 3: Mean AUROC scores (in %) and standard errors on ODDS datasets.

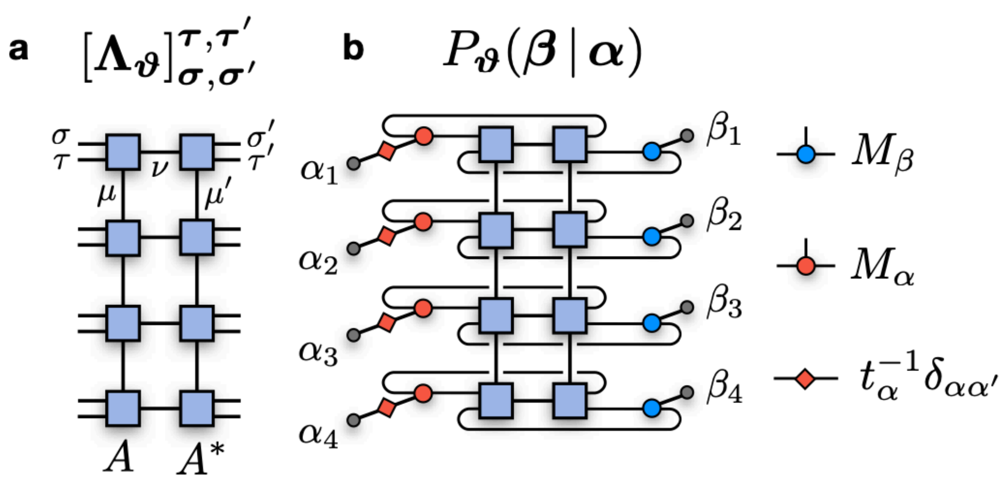
Dataset	OC-SVM	IF	GOAD	DAGMM	TNAD
Wine	**60.0**	46.0 ± 8.4	48.2 ± 24.7	51.7 ± 19.3	$\mathbf{97.3 \pm 4.5}$
Glass	**62.0**	57.2 ± 1.6	53.5 ± 13.6	52.5 ± 12.9	$\mathbf{81.8 \pm 7.3}$
Thyroid	98.8	$\mathbf{99.0 \pm 0.1}$	95.8 ± 1.3	88.8 ± 6.8	$\mathbf{99.0 \pm 0.1}$
Satellite	**79.9**	77.2 ± 0.9	60.6 ± 5.3	72.1 ± 4.7	$\mathbf{81.3 \pm 0.5}$
Forest	**97.7**	71.7 ± 2.6	64.6 ± 4.7	60.9 ± 8.9	$\mathbf{98.8 \pm 0.6}$

Results on Tabular Data

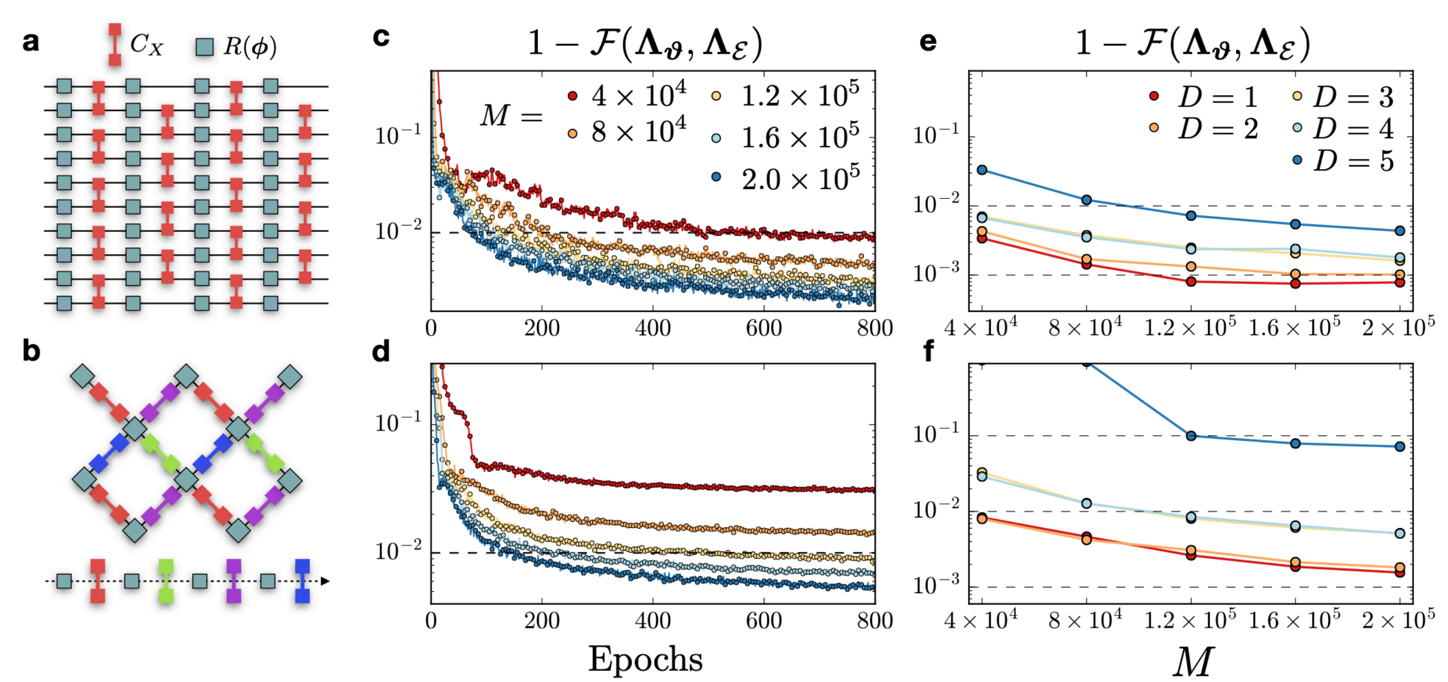
Quantum process tomography with unsupervised learning and tensor networks

Giacomo Torlai, Christopher J. Wood, Atithi Acharya, Giuseppe Carleo, Juan Carrasquilla, Leandro Aolita, arxiv: 2006.02424

- Also used squared tensor network (locally purified state)
- Deduce noisy circuit (actually CPTP map) that explains observed data
- First <u>scalable</u> approach to quantum process tomography, as far as I know



Model Architecture



Learning Random Quantum Circuits

Summary & Future Directions

Tensor networks are a natural way to parameterize interesting and powerful machine learning models

Benefits:

- Linear scaling
- Adaptive weights
- Learning data "features"
- Interpretability & theory
- Better algorithms
- Quantum computing

Much work to be done:

- best approach for various data set types
- theory of learning of tensor networks (it is possible!)
- developing new learning algorithms