Quantum Deep Learning

Daniel Persson
Department of Mathematical Sciences
Chalmers University of Technology
University of Gothenburg

Collaboration with:

@Chalmers: Robert Berman, Jimmy Aronsson (WASP PhD student)

@RISE Research Institutes of Sweden: Fredrik Ohlsson (also **@Chalmers)**

@Zenuity: Christoffer Petersson

Understanding deep learning using techniques and principles from mathematics, statistical mechanics, and quantum field theory

Daniel Persson

Department of Mathematical Sciences
Chalmers University of Technology
University of Gothenburg

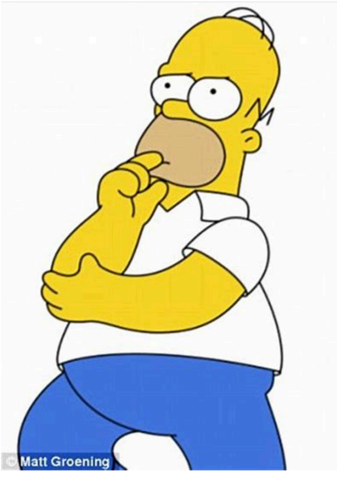
Collaboration with:

@Chalmers: **Robert Berman, Jimmy Aronsson** (WASP PhD student)

@RISE Research Institutes of Sweden: **Fredrik Ohlsson (**also **@Chalmers)**

@Zenuity: **Christoffer Petersson**

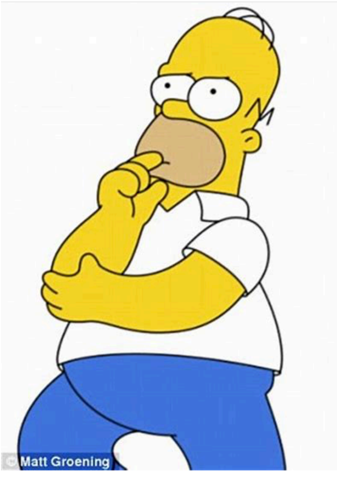
What is quantum deep learning?



What is quantum deep learning?

The answer depends on who you ask…

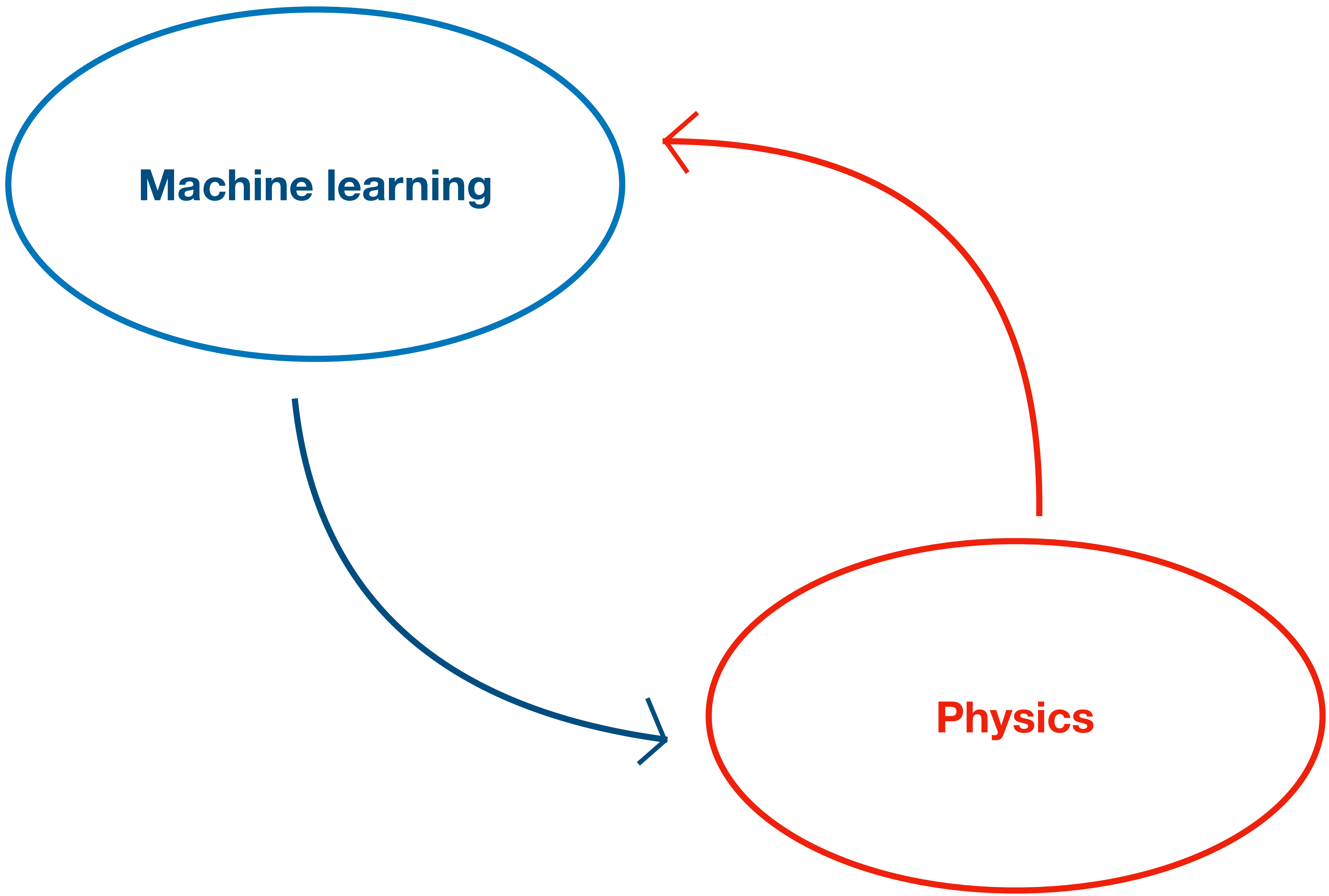
- Use (deep) neural networks to model quantum many-body systems
- Use quantum computers to train deep neural networks
- Quantum neural networks
- Use principles and techniques from quantum physics to understand deep neural networks
- …..

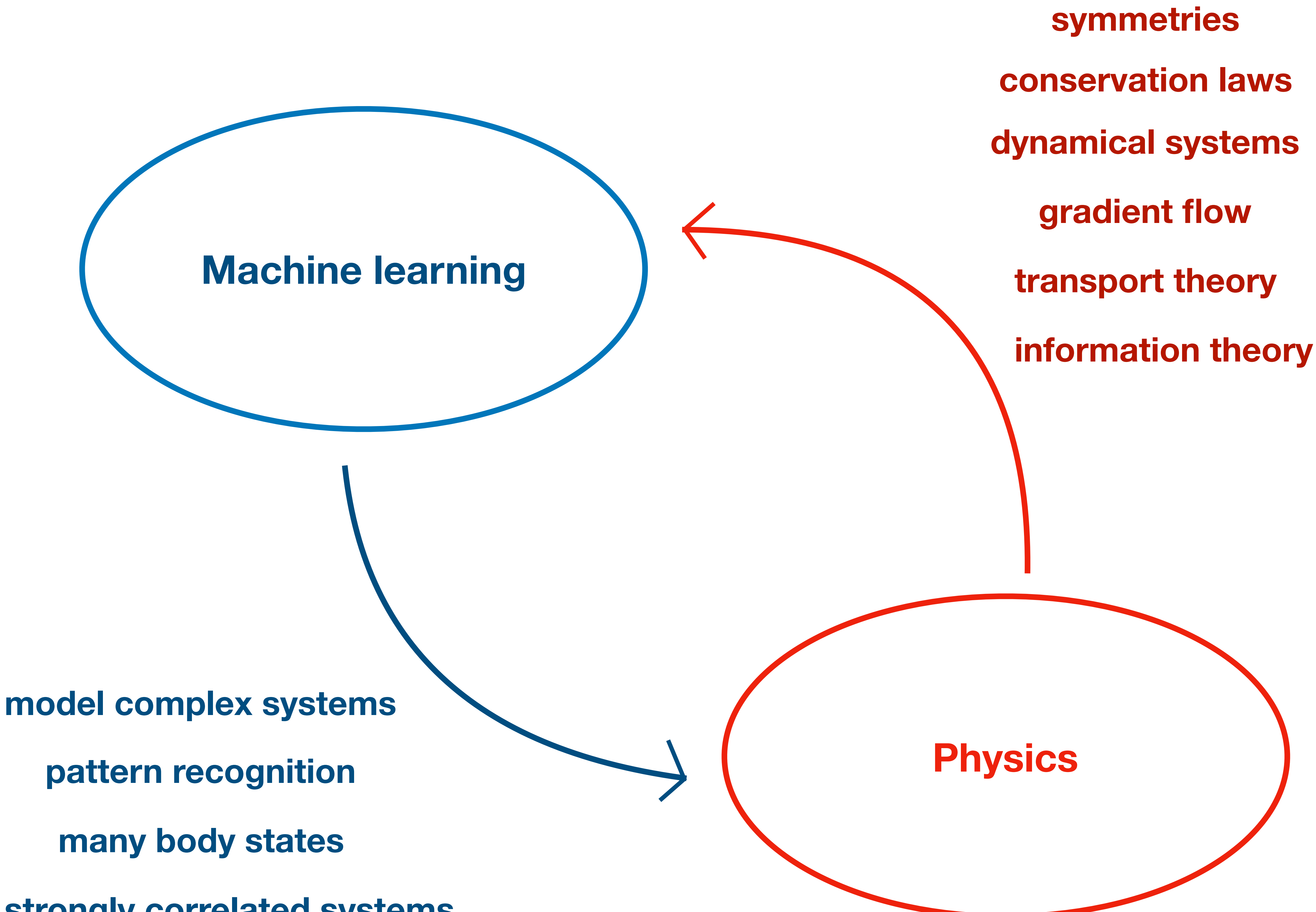


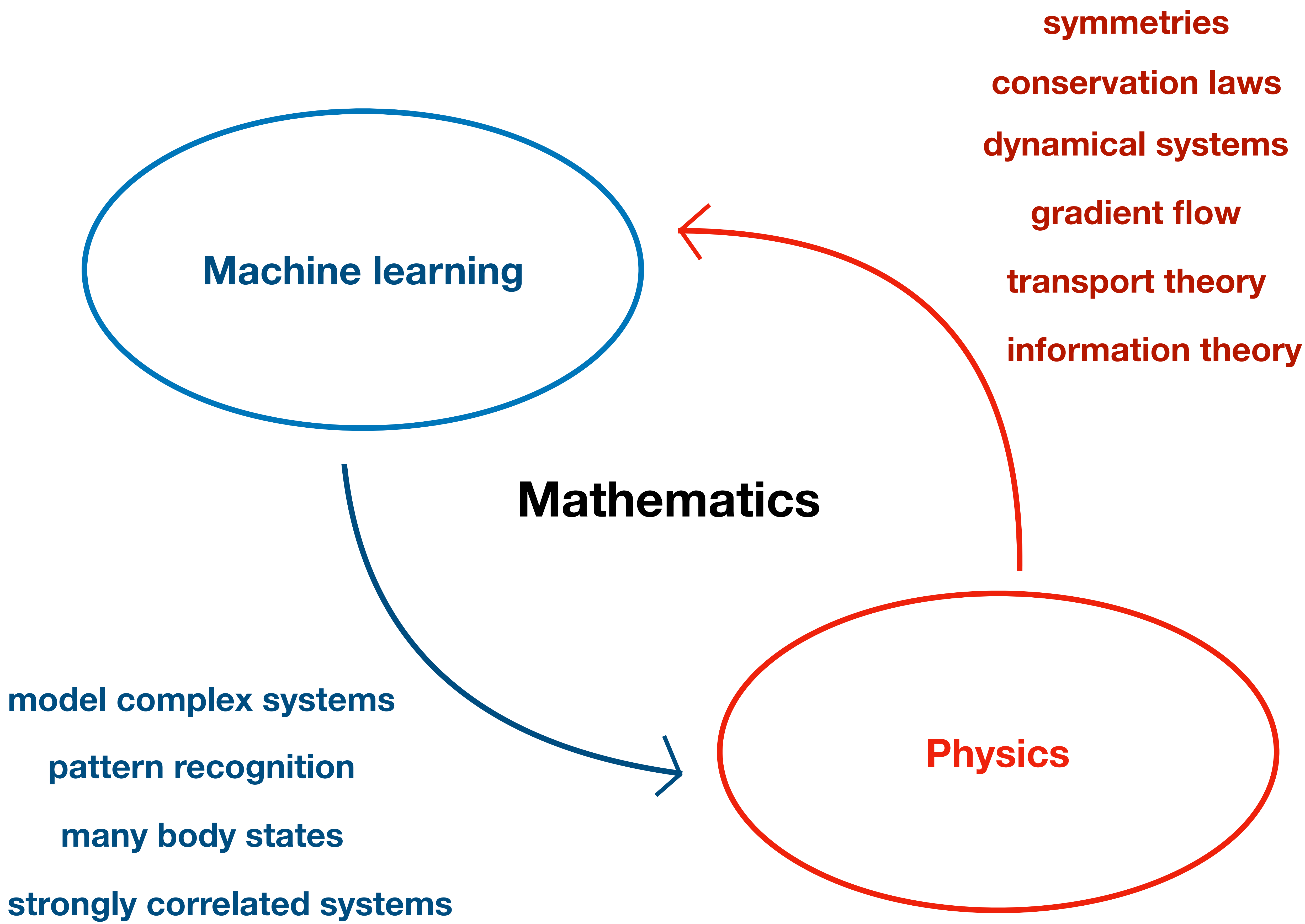
What is quantum deep learning?

The answer depends on who you ask…

- Use (deep) neural networks to model quantum many-body systems
- Use quantum computers to train deep neural networks
- Quantum neural networks
- Use principles and techniques from quantum physics to understand deep neural networks
- …..


Machine learning
Physics


Machine learning
Physics
symmetries
conservation laws
dynamical systems
gradient flow
transport theory
information theory
model complex systems
pattern recognition
many body states
strongly correlated systems


symmetries
conservation laws
dynamical systems
gradient flow
transport theory
information theory
Machine learning
Mathematics
model complex systems
pattern recognition
many body states
strongly correlated systems
Physics

Outline

1. Supervised vs. Unsupervised Learning

2. Renormalization and Boltzmann Machines

3. Deep Learning and Quantum Entanglement

4. Equivariant Convolutional Neural Networks and Gauge Theory

5. Outlook

Supervised vs. Unsupervised

Supervised learning

Training with **labelled** data

$$\{(x_1, y_1), \ldots, (x_N, y_N)\} \longrightarrow \mathcal{L}_{tot}(\lambda) = \frac{1}{N} \sum_i \mathcal{L}[y_i, \mathcal{F}(x_i; \lambda)]$$

Supervised learning

Training with **labelled** data

$$\{(x_1, y_1), \ldots, (x_N, y_N)\} \longrightarrow \mathcal{L}_{tot}(\lambda) = \frac{1}{N}\sum_i \mathcal{L}[y_i, \mathcal{F}(x_i; \lambda)]$$

data point

label

output

loss function to be minimized with respect to weights λ

Supervised learning

Training with **labelled** data

output

$$\{(x_1, y_1), \ldots, (x_N, y_N)\} \longrightarrow \mathcal{L}_{tot}(\lambda) = \frac{1}{N} \sum_i \mathcal{L}[y_i, \mathcal{F}(x_i; \lambda)]$$

data point

label

loss function to be minimized with respect to weights λ

Stochastic gradient descent

$$\lambda' = \lambda - \eta \nabla_\lambda \mathcal{L}_{tot}(\lambda)$$

Unsupervised learning

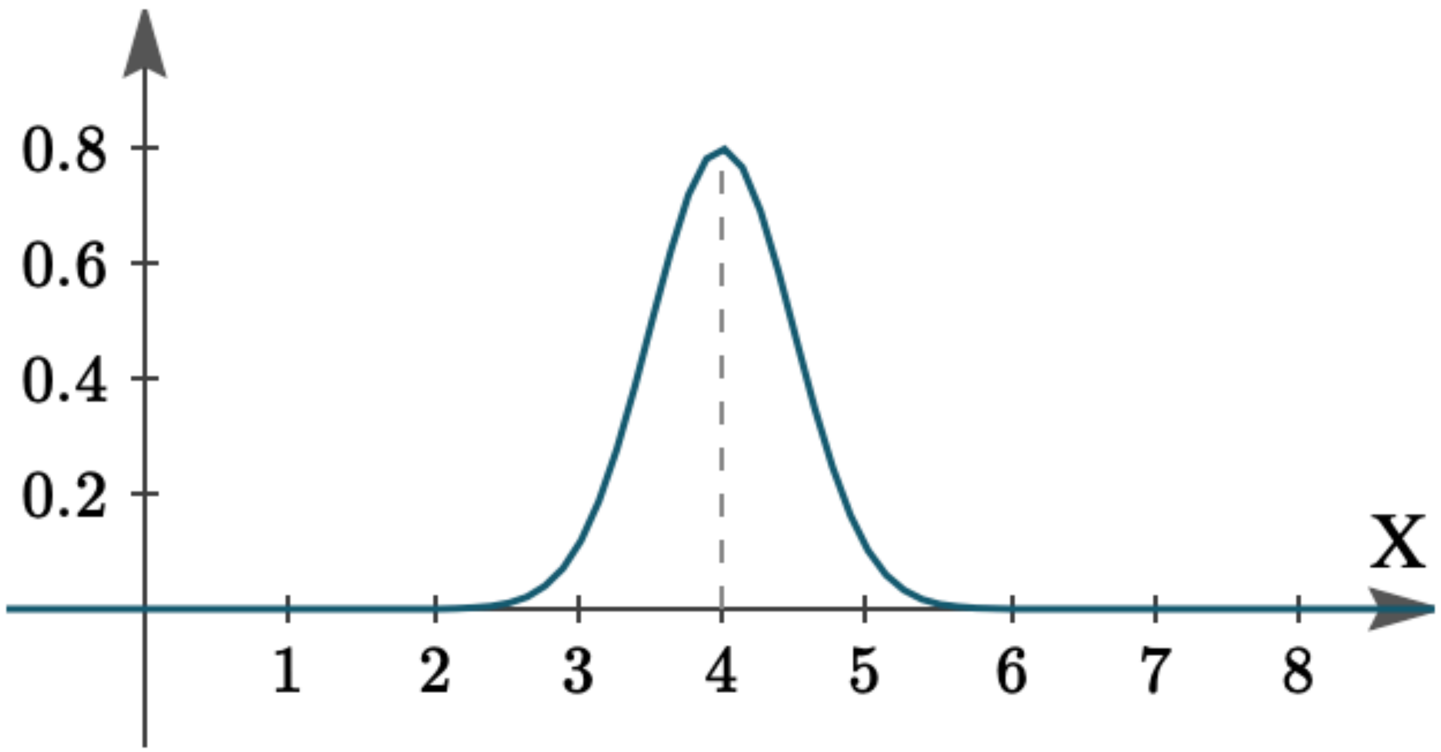
Training with **unlabelled** data

$$\{x_1, \ldots, x_N\}$$

data points drawn from
unknown distribution $P(x)$

Goal: Find a model distribution

$$p_\lambda(x) \cong P(x)$$



Unsupervised learning

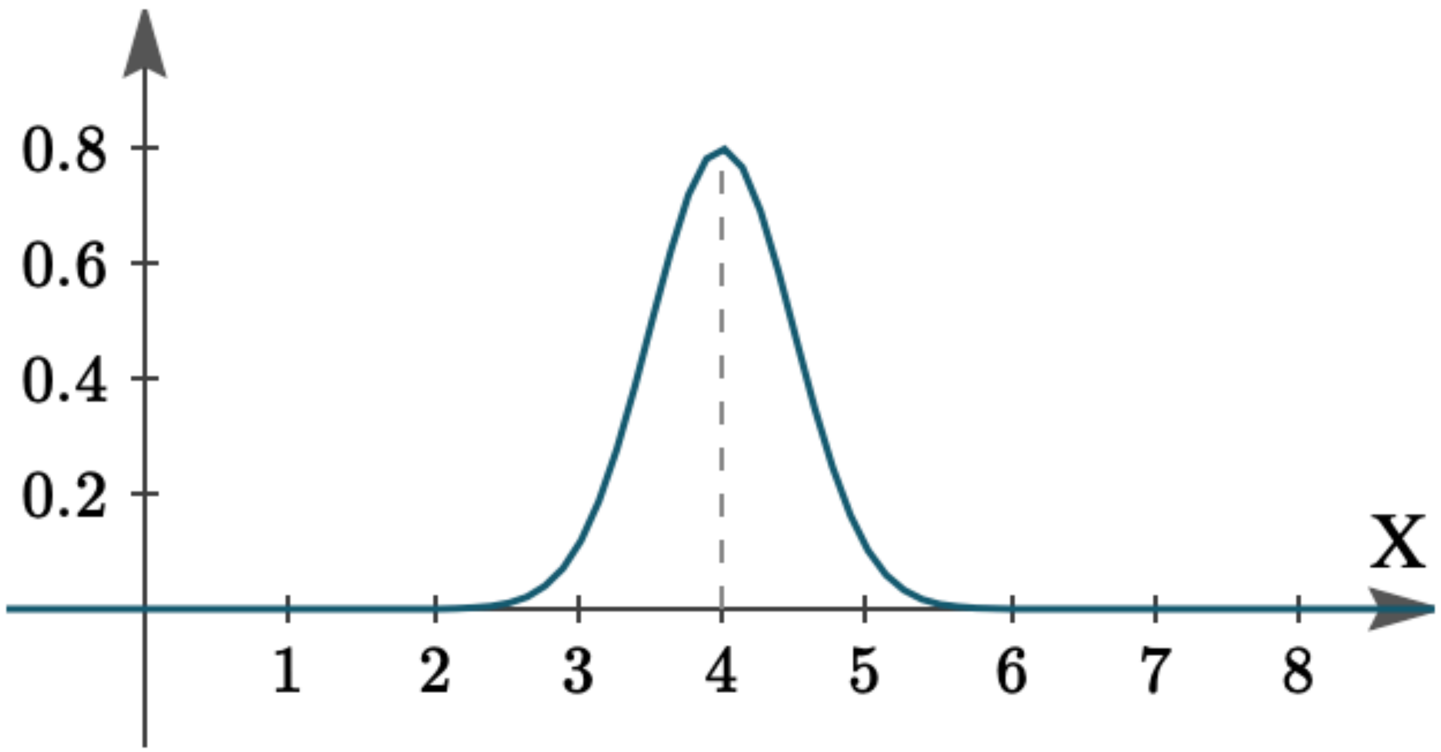
Training with **unlabelled** data

$$\{x_1, \ldots, x_N\}$$

data points drawn from
unknown distribution $P(x)$

Goal: Find a model distribution

$$p_\lambda(x) \cong P(x)$$



Optimization problem:

$$\text{argmin}_\lambda \mathcal{L}[P, p_\lambda]$$

Unsupervised learning

Training with **unlabelled** data

$$\{x_1, \ldots, x_N\}$$

data points drawn from
unknown distribution $P(x)$

Goal: Find a model distribution

$$p_\lambda(x) \cong P(x)$$

Optimization problem:

$$\mathrm{argmin}_\lambda \mathcal{L}[P, p_\lambda]$$

Once learned we can **draw new samples** from the model distribution

[Picture from thispersondoesnotexist.com]

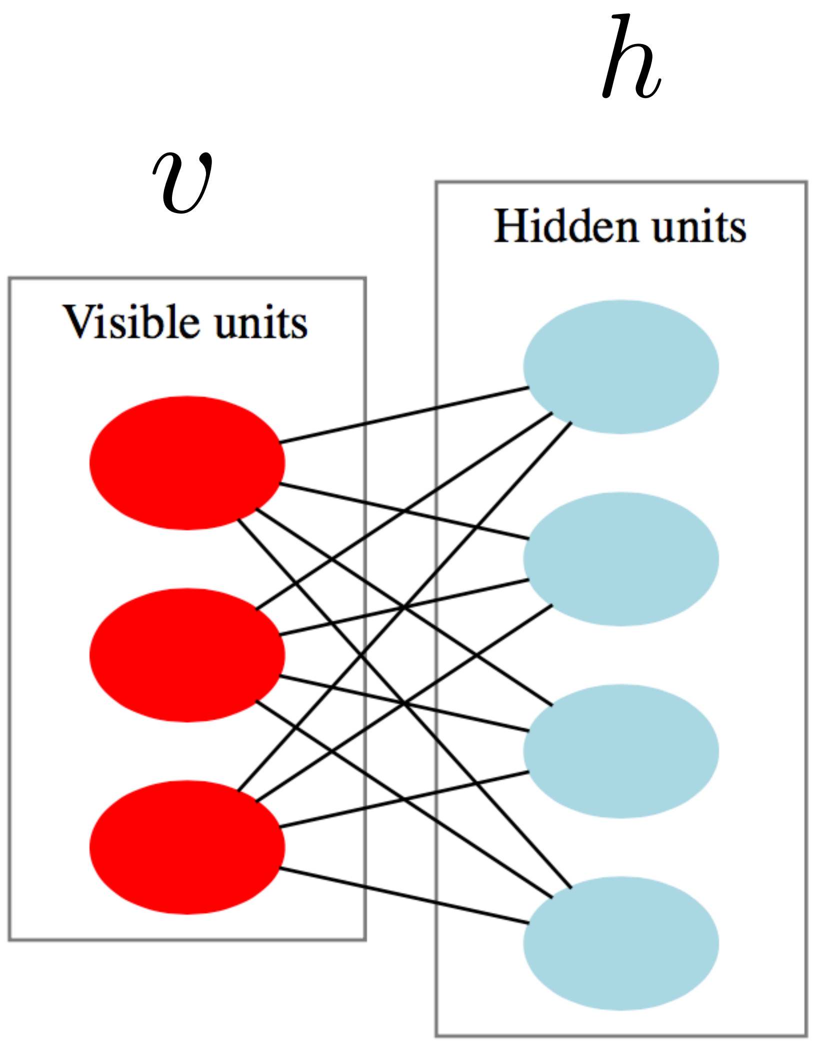
Picture generated using a GAN

Boltzmann Machines
& Renormalization

Restricted Boltzmann Machines [Hinton]

Neural Network with **no intralayer** interactions

Neurons form a **bipartite graph**



Joint probability distribution:

$$p_\lambda(v,h) = \frac{1}{\mathcal{Z}} e^{-E(v,h)}$$

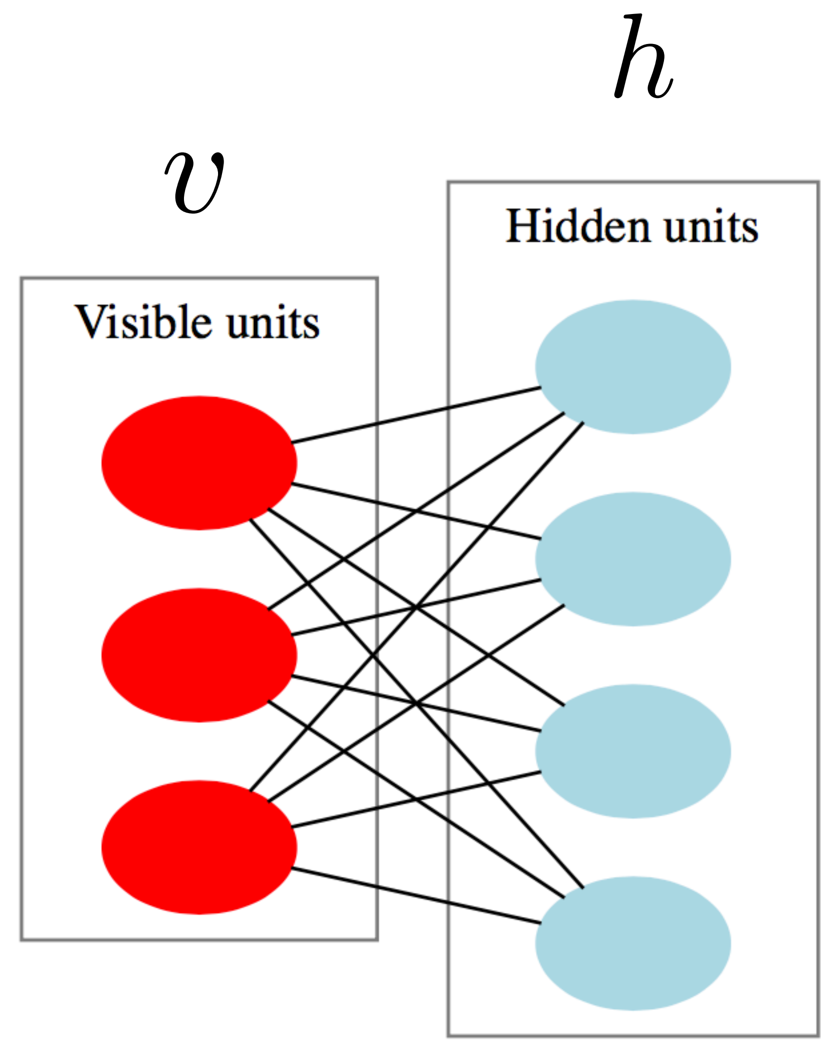
True distribution:

$P(v)$ **(unknown)**

Restricted Boltzmann Machines [Hinton]

Neural Network with **no intralayer** interactions

Neurons form a **bipartite graph**



Joint probability distribution:

$$p_\lambda(v,h) = \frac{1}{\mathcal{Z}} e^{-E(v,h)}$$

True distribution:

$P(v)$ **(unknown)**

Energy function:

$$E(v,h) = v^T W h + av + bh$$

Variational parameters (weights):

$$\lambda = \{W, a, b\}$$

Training

Fix the parameters λ to **minimize** the Kullback-Leibler divergence

$$D_{KL}(P||p_\lambda) = -\sum_v P(v) \log\left(\frac{p_\lambda(v)}{P(v)}\right)$$

Training

Fix the parameters λ to **minimize** the Kullback-Leibler divergence

$$D_{KL}(P||p_\lambda) = -\sum_v P(v) \log\left(\frac{p_\lambda(v)}{P(v)}\right)$$

$$p_\lambda(v) = \sum_h p_\lambda(v,h)$$

marginal distribution

Training

Fix the parameters λ to **minimize** the Kullback-Leibler divergence

$$D_{KL}(P||p_\lambda) = -\sum_v P(v)\log\left(\frac{p_\lambda(v)}{P(v)}\right)$$

Unsupervised learning: The distribution $p_\lambda(v)$ is used to model the true distribution $P(v)$ of the training data set

➡ Once learned we can draw **new samples** from our model distribution

Training

Fix the parameters λ to **minimize** the Kullback-Leibler divergence

$$D_{KL}(P||p_\lambda) = -\sum_v P(v) \log\left(\frac{p_\lambda(v)}{P(v)}\right)$$

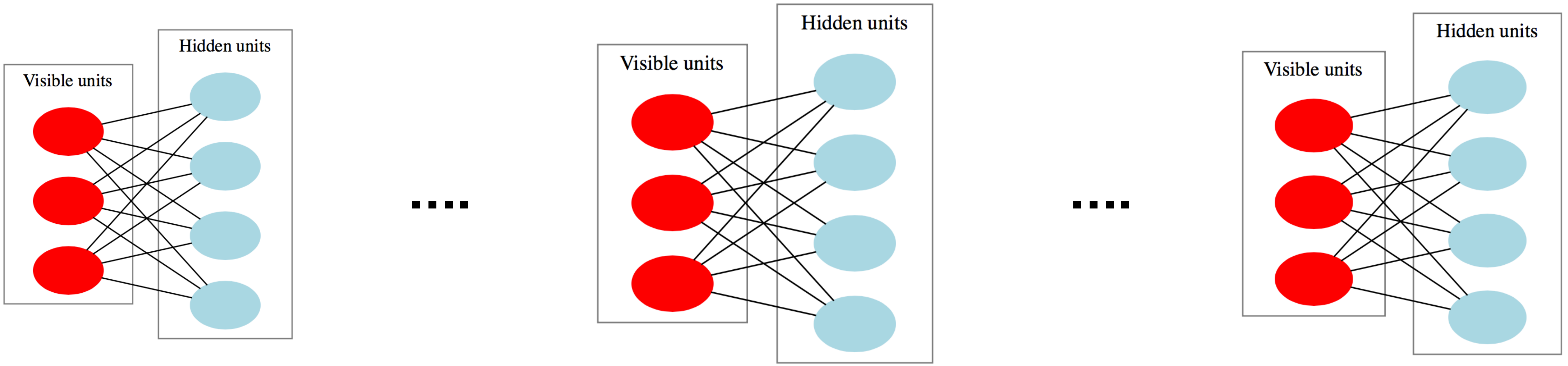
Unsupervised learning: The distribution $p_\lambda(v)$ is used to model the true distribution $P(v)$ of the training data set

➡ Once learned we can draw **new samples** from our model distribution

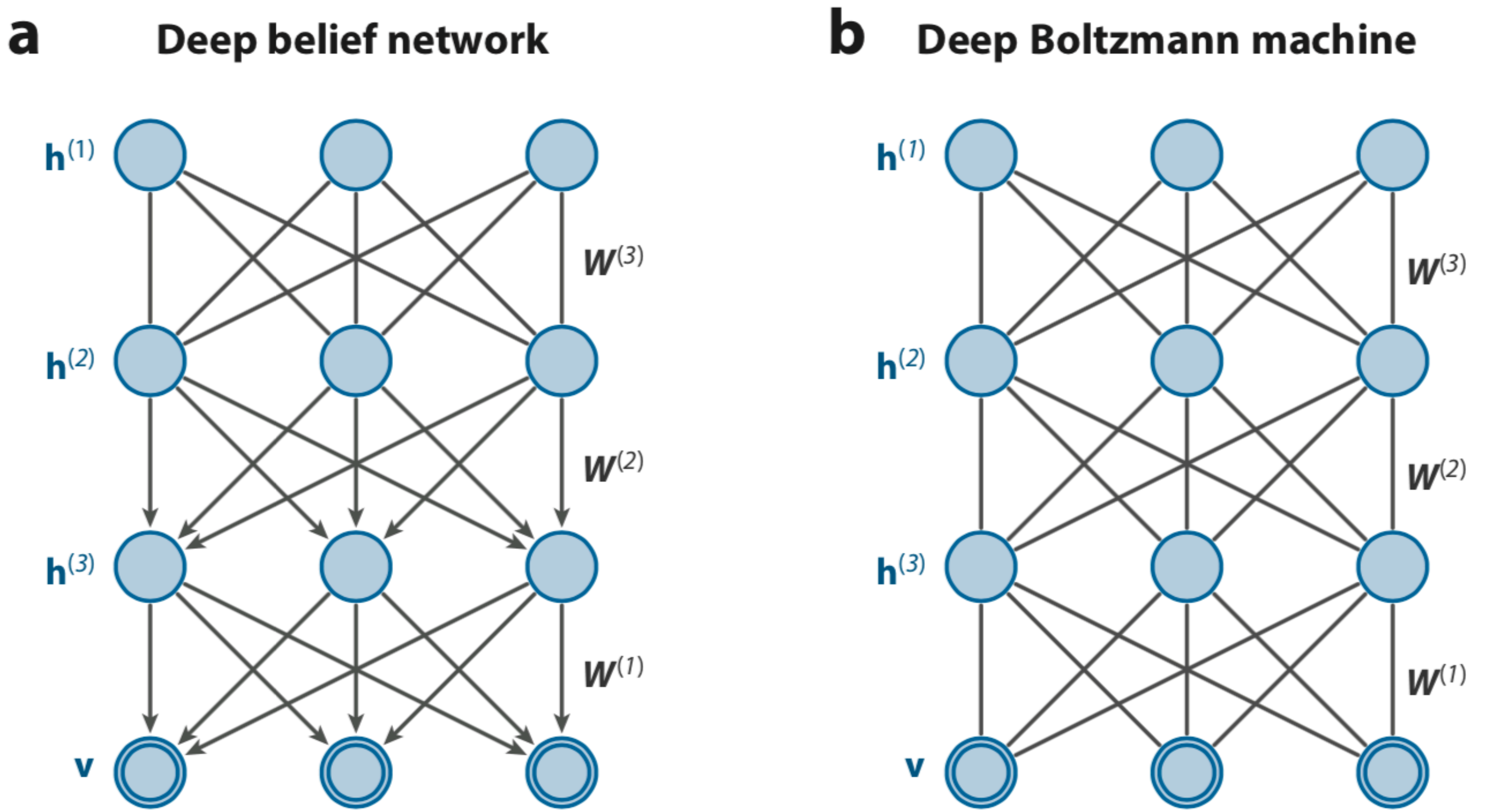
Representability theorem [Le Roux, Bengio]:
A restricted Boltzmann machine can model any discrete probability distribution given sufficiently many hidden units.

Deep Boltzmann Machine [Hinton]

Form a **deep neural network** by stacking these on top of each other



Form a **deep neural network** by stacking these on top of each other



[Figure from Salakhutdinov]

Renormalization group flow of neural networks

[Mehta, Schwab][Benyi][de Mello Koch, de Mello Koch, Cheng]

Key idea: The process of training a neural network implements a **renormalization group flow** on the space of parameters

$$\rightarrow p_{\lambda} \rightarrow p_{\lambda'} \rightarrow p_{\lambda''} \rightarrow$$

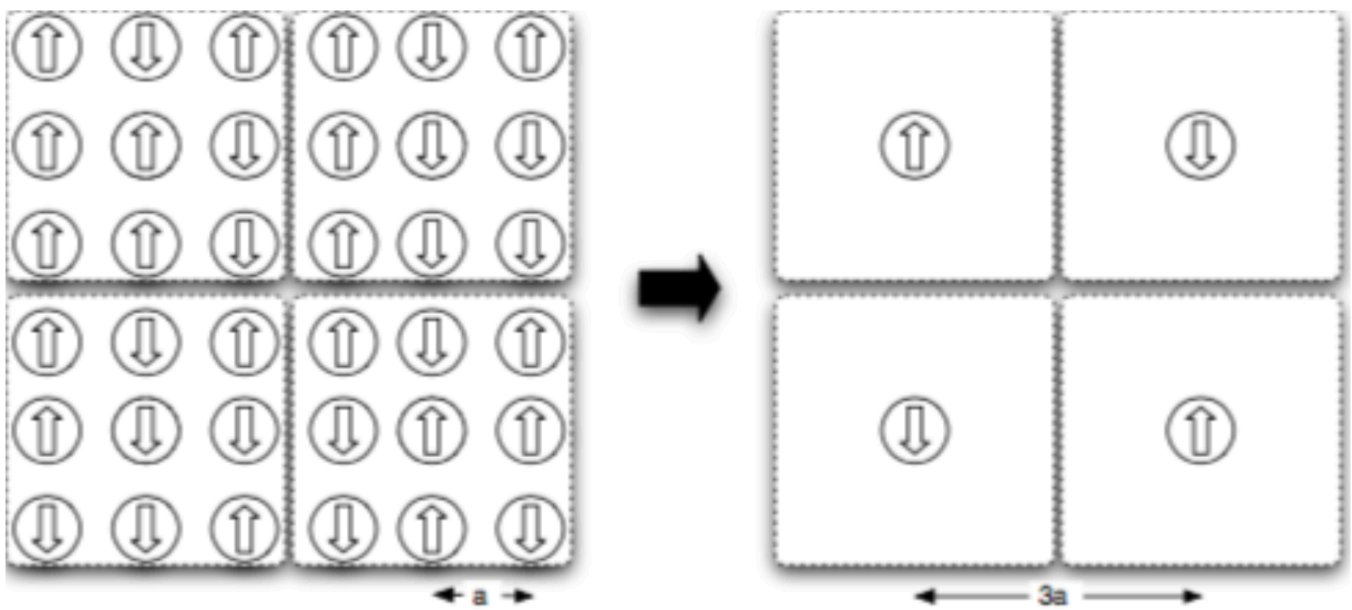
Renormalization group flow of neural networks

[Mehta, Schwab][Benyi][de Mello Koch, de Mello Koch, Cheng]

Key idea: The process of training a neural network implements a **renormalization group flow** on the space of parameters

{microscopic} $\rightarrow p_\lambda \rightarrow p_{\lambda'} \rightarrow p_{\lambda''} \rightarrow$ {macroscopic}

Statistical physics: Different microscopic theories flow to one macroscopic theory.



block-spin renormalization

[Kadanoff]

Suggests some kind of universality property of neural networks

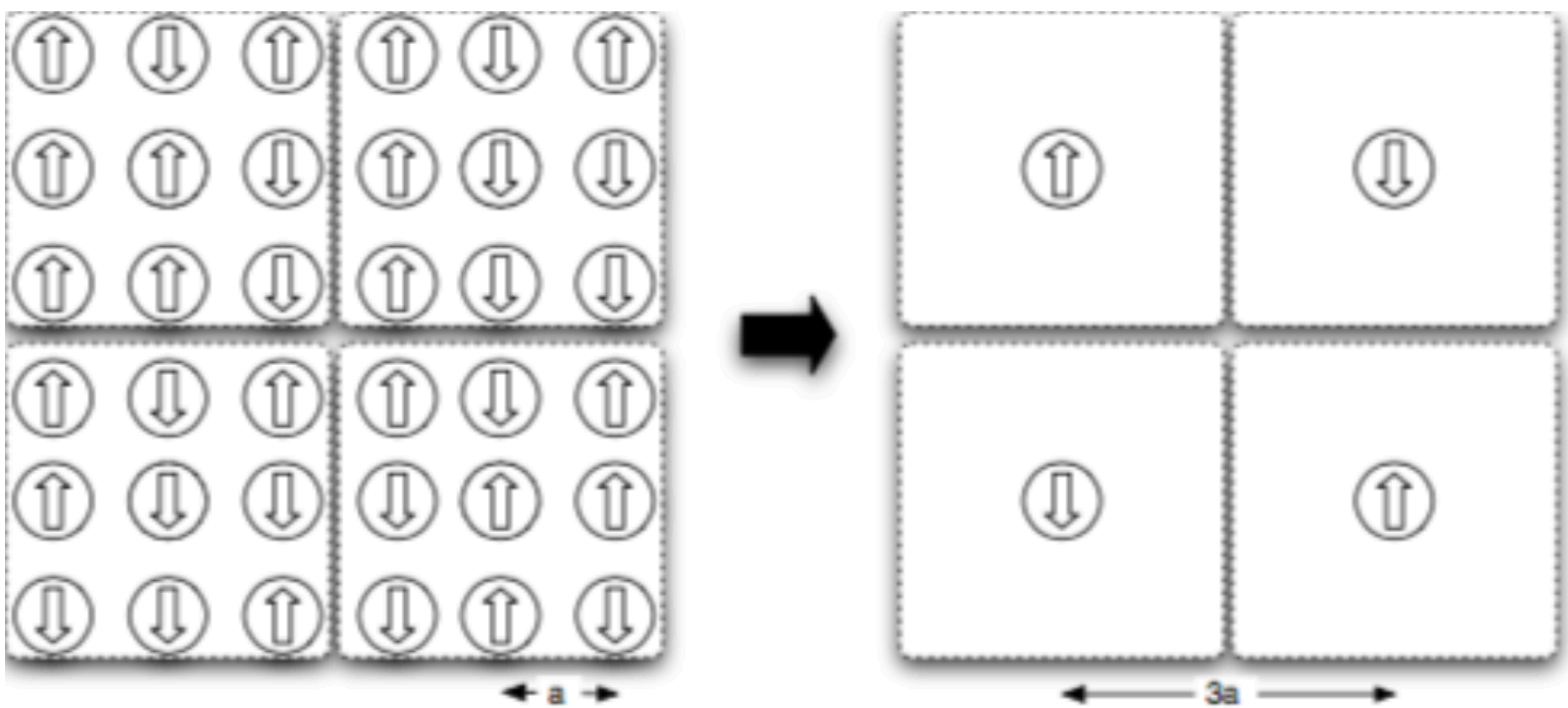
Block-spin renormalization [Kadanoff]

Renormalization sets the scale beyond which new physics occur and below which the model we study is a good effective model of the physics.

Block-spin renormalization [Kadanoff]

Renormalization sets the scale beyond which new physics occur and below which the model we study is a good effective model of the physics.

We can think of renormalization as a **flow** to larger and larger scales



Statistical mechanical setup

state variables (e.g. spins) $\{s_i\}$ $i = 1, \ldots, N$

coupling constants $\{J_\mu\}$ (strength of interactions)

partition function $$\mathcal{Z}(\{J_\mu\}) = \sum_{\{s_i\}} e^{-H(\{s_i\},\{J_\mu\})}$$

Statistical mechanical setup

state variables (e.g. spins) $\{s_i\}$ $i = 1, \ldots, N$

coupling constants $\{J_\mu\}$ (strength of interactions)

partition function $$\mathcal{Z}(\{J_\mu\}) = \sum_{\{s_i\}} e^{-H(\{s_i\},\{J_\mu\})}$$

Hamiltonian

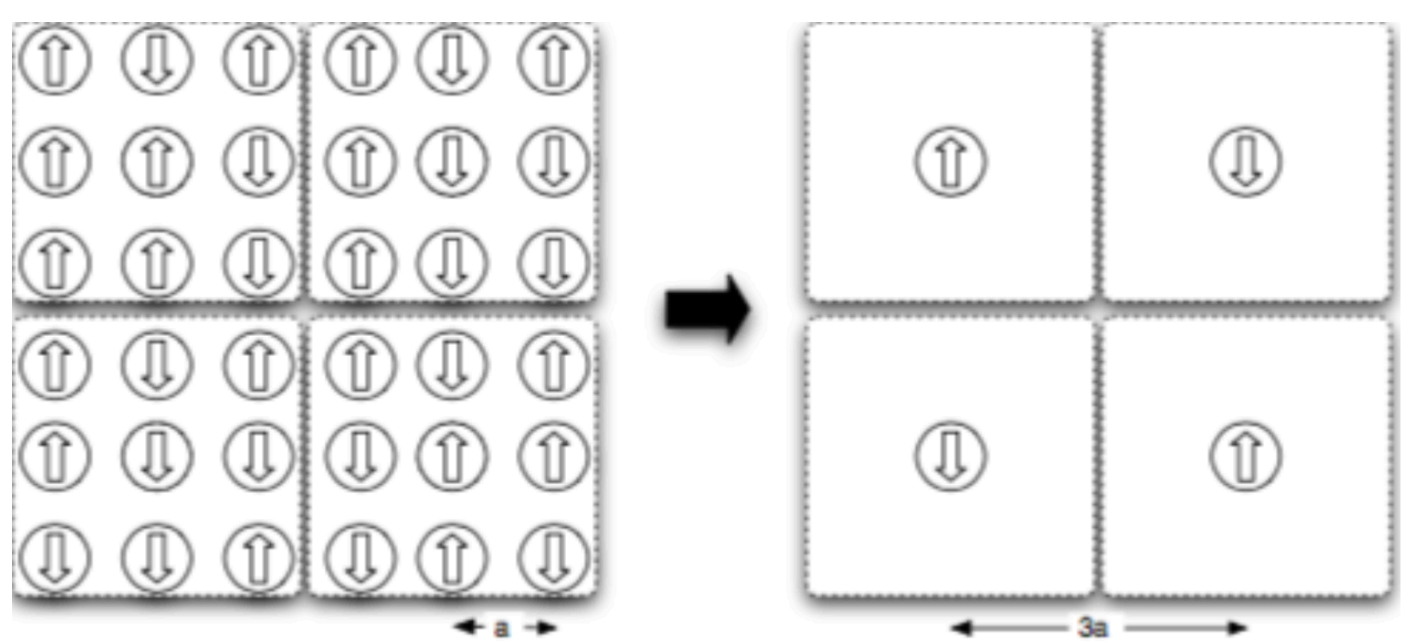
Statistical mechanical setup

state variables (e.g. spins) $\{s_i\}$ $i = 1, \ldots, N$

coupling constants $\{J_\mu\}$ (strength of interactions)

partition function $$\mathcal{Z}(\{J_\mu\}) = \sum_{\{s_i\}} e^{-H(\{s_i\},\{J_\mu\})}$$

Replace: $\{s_i\} \longrightarrow \{s'_j\}$ **block variables** $j = 1, \ldots, N' < N$



Statistical mechanical setup

state variables (e.g. spins) $\{s_i\}$ $i = 1, \ldots, N$

coupling constants $\{J_\mu\}$ (strength of interactions)

partition function $$\mathcal{Z}(\{J_\mu\}) = \sum_{\{s_i\}} e^{-H(\{s_i\},\{J_\mu\})}$$

Replace: $\{s_i\} \longrightarrow \{s'_j\}$ block variables $j = 1, \ldots, N' < N$

If there exists new couplings $\{J'_\nu\}$ such that the Hamiltonian of the new system is of the same form $H(\{s'_j\}, \{J'_\nu\})$ this is a **renormalization transf**.

Iterating this process gives a **renormalization group flow**

$$H(\{s\},\{J\}) \longrightarrow H(\{s'\},\{J'\}) \longrightarrow H(\{s''\},\{J''\}) \longrightarrow \cdots$$

This is a **coarse graining** of the degrees of freedom

Iterating this process gives a **renormalization group flow**

$$H(\{s\},\{J\}) \longrightarrow H(\{s'\},\{J'\}) \longrightarrow H(\{s''\},\{J''\}) \longrightarrow \cdots$$

This is a **coarse graining** of the degrees of freedom

Each transformation is implemented via a **RG-function** $T(\{s\},\{s'\})$

by **"integrating out"** the original state variables

$$e^{-H(\{s'_j\})} = \sum_i e^{T(\{s_i\},\{s'_j\})-H(\{s_i\})}$$

Relation with renormalization

(following Mehta & Schwab)

Key observation: The map between visible and hidden units in a RBM is equivalent to the renormalization group flow

$$H(\{s\},\{J\}) \longrightarrow H(\{s'\},\{J'\}) \longrightarrow H(\{s''\},\{J''\}) \longrightarrow \cdots$$

Relation with renormalization

(following Mehta & Schwab)

Key observation: The map between visible and hidden units in a RBM is equivalent to the renormalization group flow

$$H(\{s\},\{J\}) \longrightarrow H(\{s'\},\{J'\}) \longrightarrow H(\{s''\},\{J''\}) \longrightarrow \cdots$$

Renormalization group flow		$\sim$	**Deep Boltzmann Machine**	
state variables	$\{s\}$		$\{v\}$	visible units
coarse-grained variables	$\{s'\}$		$\{h\}$	hidden units
RG-transform	$T(\{v\},\{h\})$		$E(\{v\},\{h\})$	energy function

Relation with renormalization

(following Mehta & Schwab)

Key observation: The map between visible and hidden units in a RBM is equivalent to the renormalization group flow

$$H(\{s\},\{J\}) \longrightarrow H(\{s'\},\{J'\}) \longrightarrow H(\{s''\},\{J''\}) \longrightarrow \cdots$$

Renormalization group flow		$\sim$		**Deep Boltzmann Machine**
state variables	$\{s\}$		$\{v\}$	visible units
coarse-grained variables	$\{s'\}$		$\{h\}$	hidden units

$$H(\{v\}) - T(\{v\},\{h\}) = E(\{v\},\{h\})$$

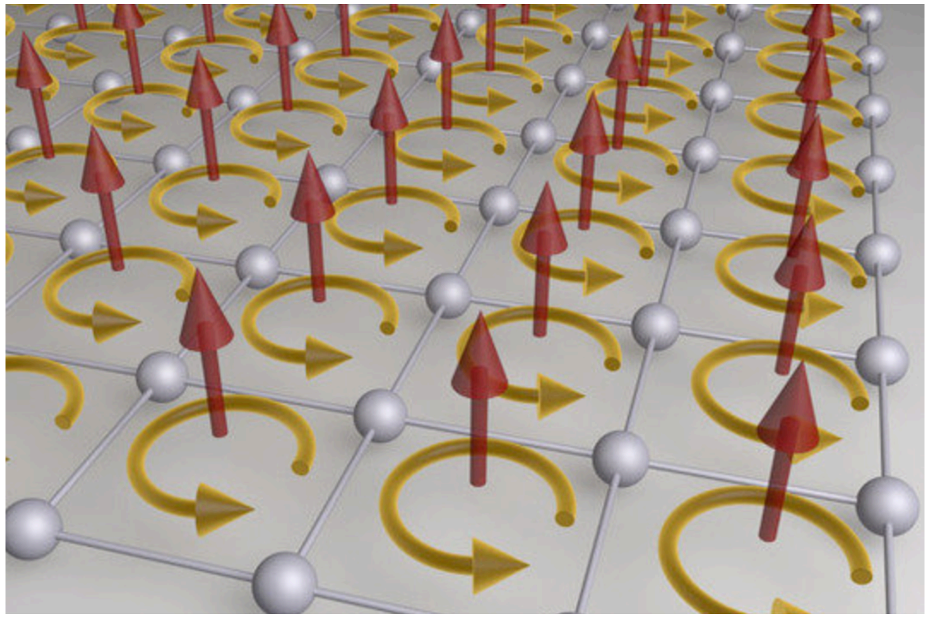
Deep Learning Quantum Many Body Physics

Quantum many-body dynamics

$\mathcal{H}$

interacting **Hamiltonian**

microscopic degrees of freedom
(e.g. electrons, cold atoms…)



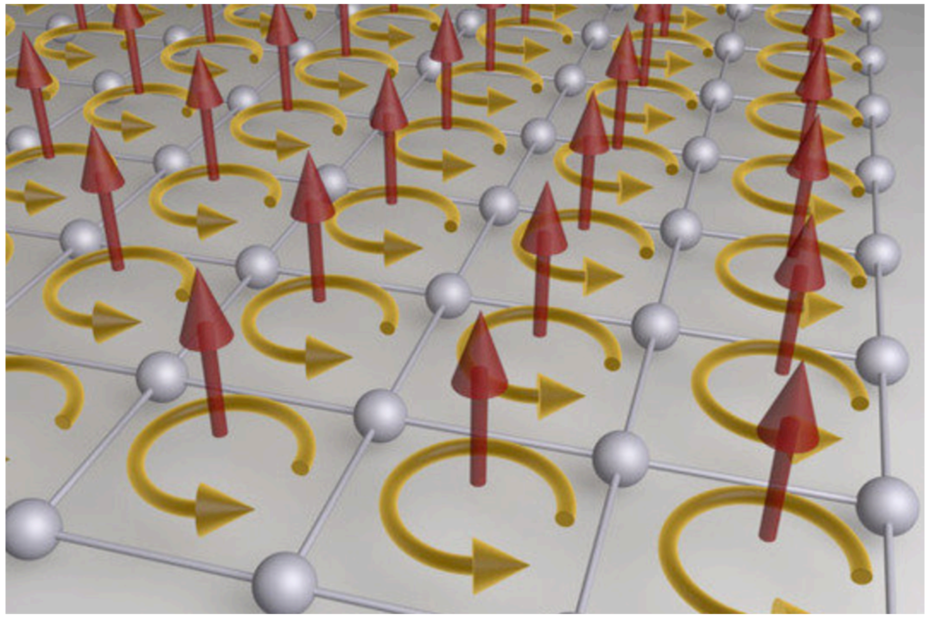
[Image: https://www.mpq.mpg.de/4743113/quasim]

Quantum many-body dynamics

$\mathcal{H}$

interacting **Hamiltonian**

microscopic degrees of freedom (e.g. electrons, cold atoms…)



[Image: https://www.mpq.mpg.de/4743113/quasim]

Ground state energy:

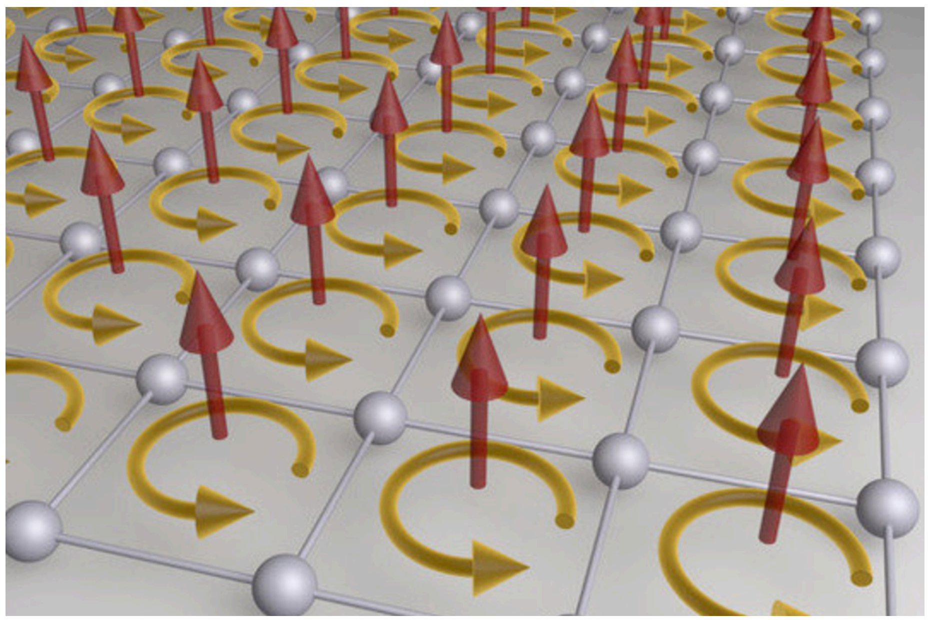
$$\mathcal{H}|\Psi\rangle = E|\Psi\rangle$$

Quantum many-body dynamics

$\mathcal{H}$

interacting **Hamiltonian**

microscopic degrees of freedom
(e.g. electrons, cold atoms…)



[Image: https://www.mpq.mpg.de/4743113/quasim]

To find the dynamics, solve the
Schrödinger equation

$$\frac{d}{dt}|\Psi\rangle = -i\mathcal{H}|\Psi\rangle$$

Ground state energy:

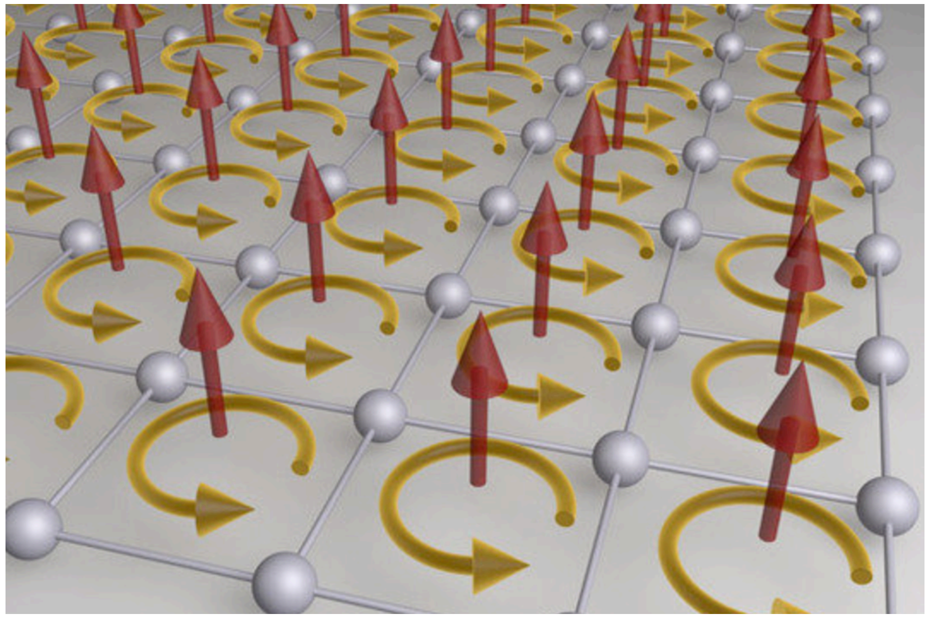
$$\mathcal{H}|\Psi\rangle = E|\Psi\rangle$$

Quantum many-body dynamics

$$\mathcal{H}$$

interacting **Hamiltonian**

microscopic degrees of freedom (e.g. electrons, cold atoms…)



[Image: https://www.mpq.mpg.de/4743113/quasim]

To find the dynamics, solve the **Schrödinger equation**

$$\frac{d}{dt}|\Psi\rangle = -i\mathcal{H}|\Psi\rangle$$

Ground state energy:

$$\mathcal{H}|\Psi\rangle = E|\Psi\rangle$$

Very difficult problem!

Quantum Monte Carlo, Tensor Networks…

Using neural networks to model quantum dynamics

[Carleo & Troyer, Science 355, 602 (2017)]

Key idea: Model the quantum wave function using a restricted Boltzmann machine

$$\Psi_\lambda(v) = \sum_h e^{v^T W h + a^T v + b^T h}$$

marginal distribution

Using neural networks to model quantum dynamics

[Carleo & Troyer, Science 355, 602 (2017)]

Key idea: Model the quantum wave function using a restricted Boltzmann machine

$$\Psi_\lambda(v) = \sum_h e^{v^T W h + a^T v + b^T h}$$

marginal distribution

Ex: Find the ground state by
minimizing the energy

$$E_\lambda = \frac{\langle \Psi_\lambda | \mathcal{H} | \Psi_\lambda \rangle}{\langle \Psi_\lambda | \Psi_\lambda \rangle}$$

Using neural networks to model quantum dynamics

[Carleo & Troyer, Science 355, 602 (2017)]

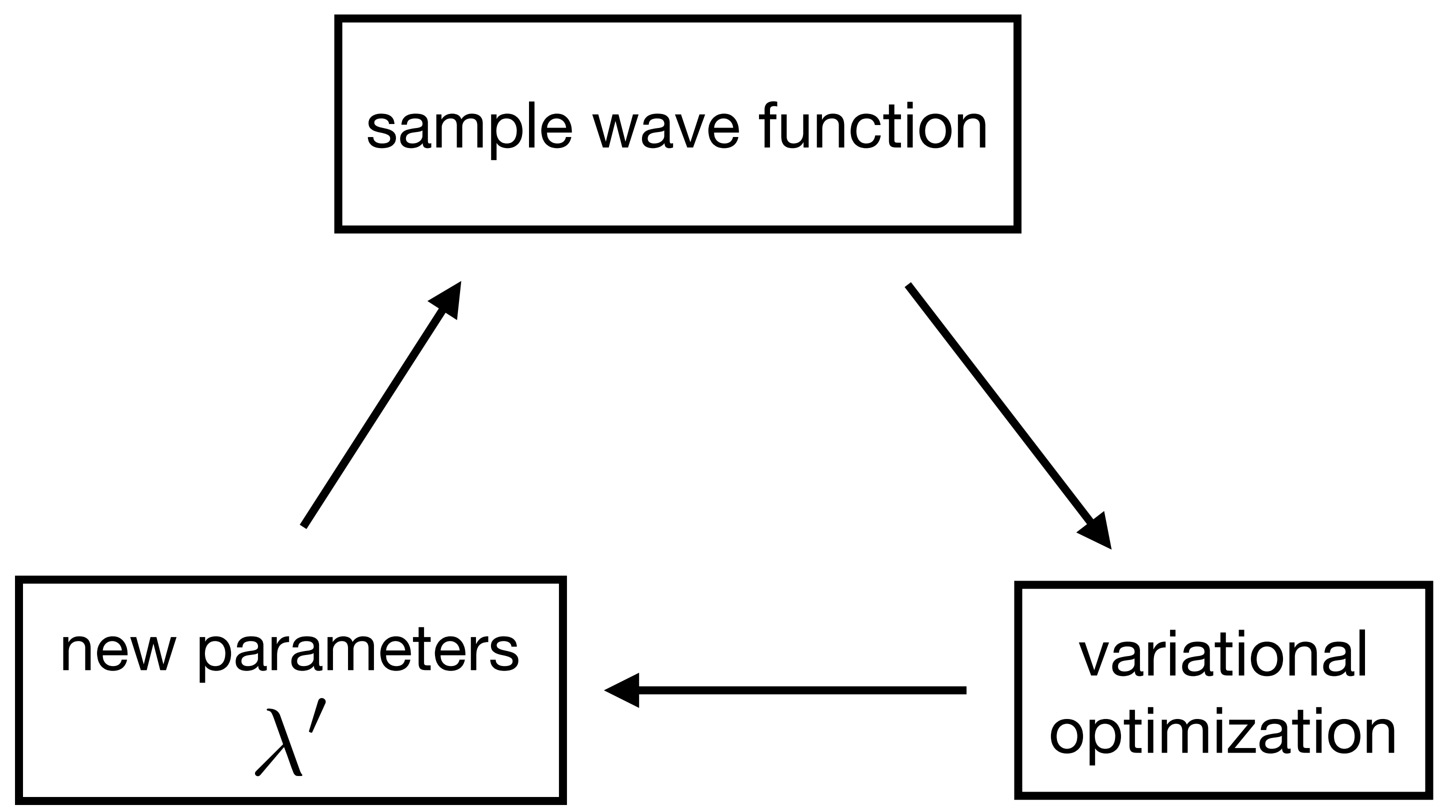
Key idea: Model the quantum wave function using a restricted Boltzmann machine

$$\Psi_\lambda(v) = \sum_h e^{v^T W h + a^T v + b^T h}$$

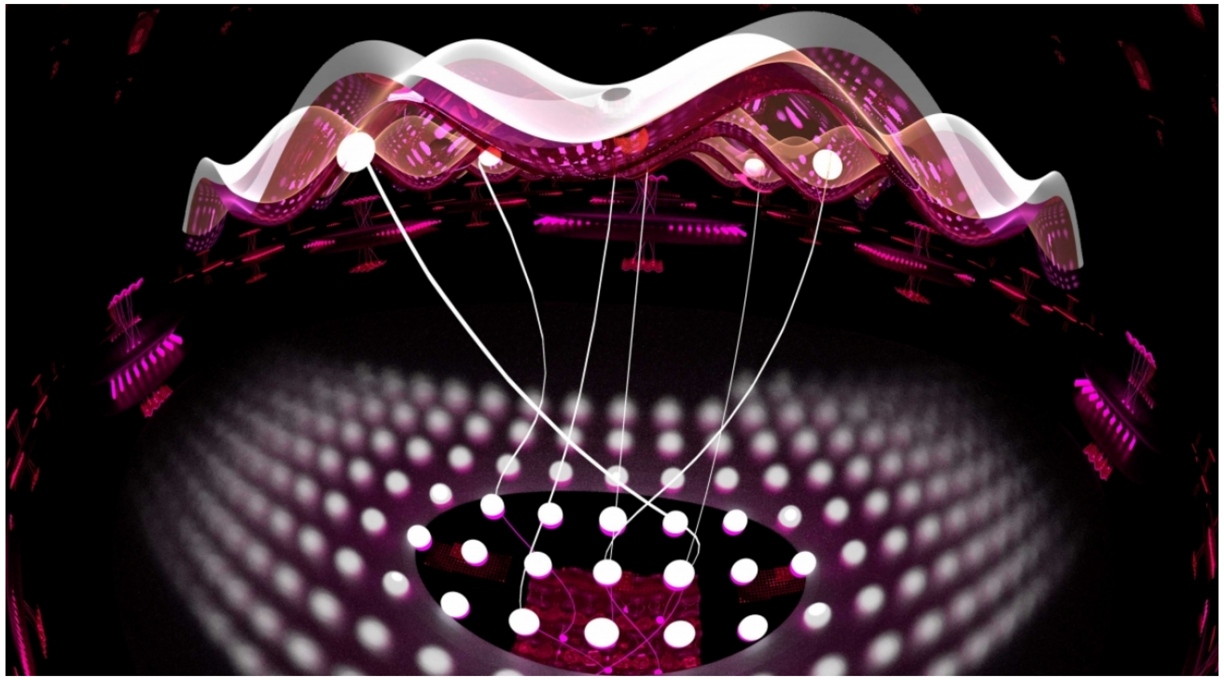
marginal distribution

Ex: Find the ground state by **minimizing the energy**

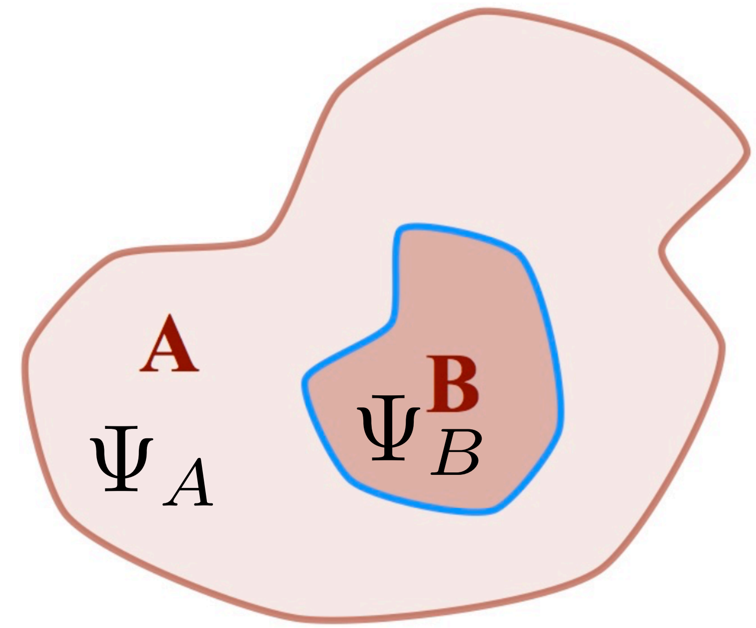
$$E_\lambda = \frac{\langle \Psi_\lambda | \mathcal{H} | \Psi_\lambda \rangle}{\langle \Psi_\lambda | \Psi_\lambda \rangle}$$



Entanglement entropy as a measure of expressiveness

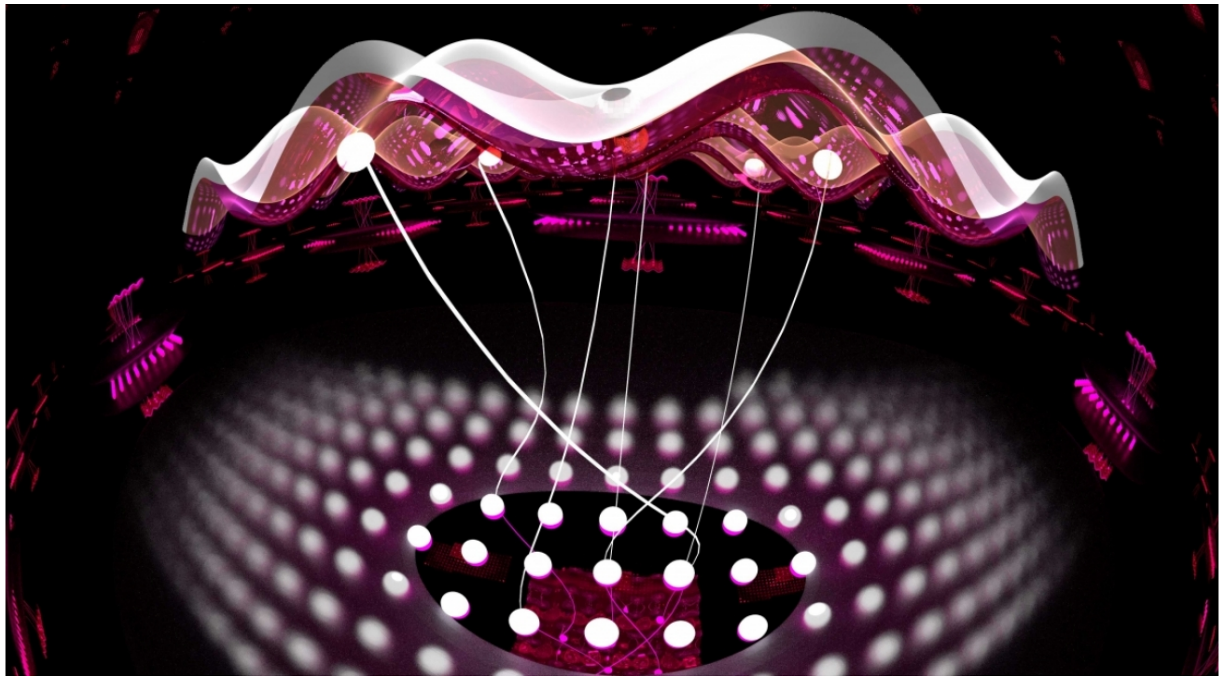


[Image: E. Edwards, Joint Quantum Institute]
(https://phys.org/news/2017-06-neural-networks-quantum-entanglement.html)

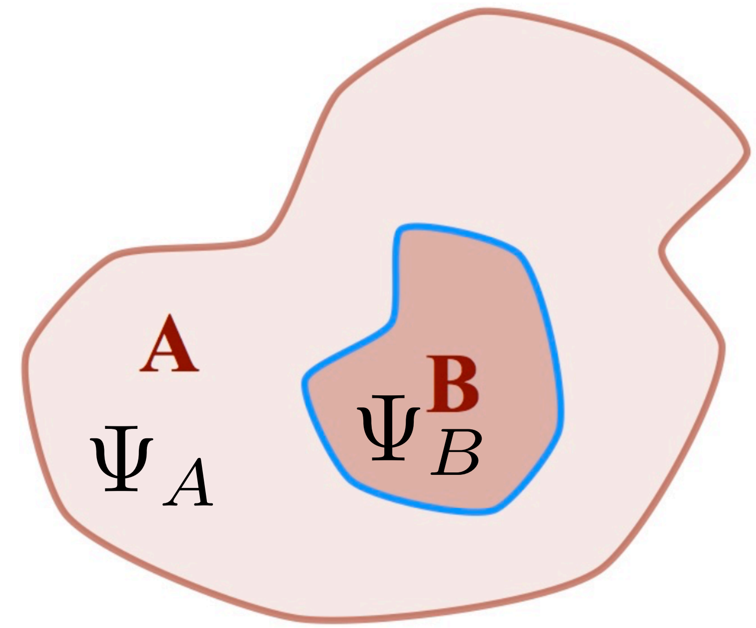


A measure of how “entangled” two quantum states are.

Entanglement entropy as a measure of expressiveness



[Image: E. Edwards, Joint Quantum Institute]
(https://phys.org/news/2017-06-neural-networks-quantum-entanglement.html)



A measure of how “entangled” two quantum states are.

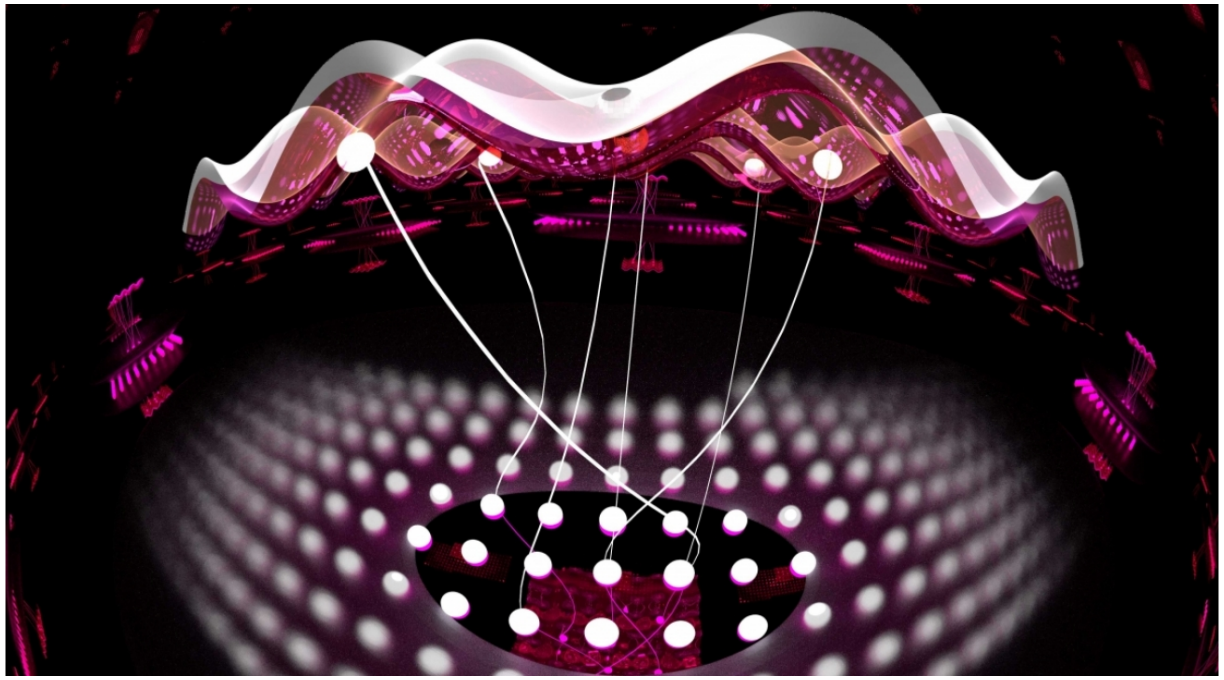
$$\rho_A(|\Psi\rangle) = \mathrm{Tr}_B(|\Psi\rangle\langle\Psi|)$$

reduced density matrix

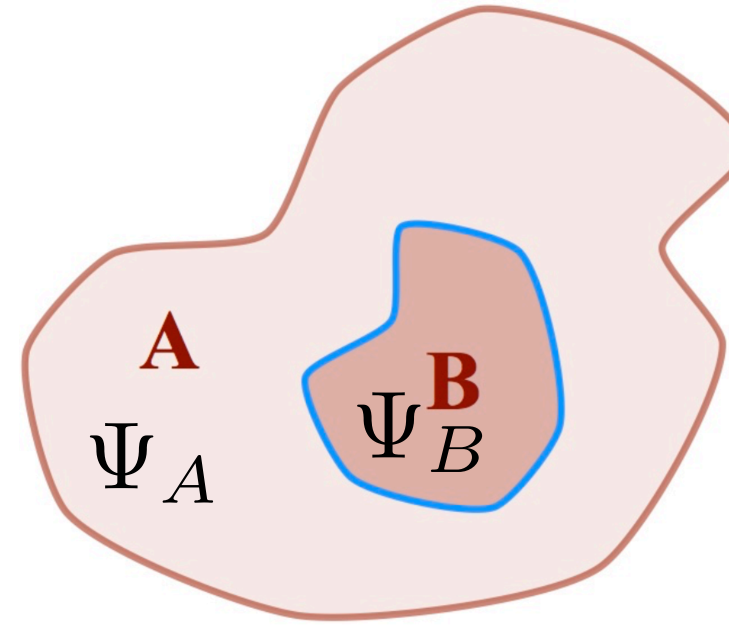
$$S(A) = -\mathrm{Tr}(\rho_A \log \rho_A)$$

Entanglement entropy
(Von Neumann entropy)

Entanglement entropy as a measure of expressiveness



[Image: E. Edwards, Joint Quantum Institute]
(https://phys.org/news/2017-06-neural-networks-quantum-entanglement.html)



A measure of how "entangled" two quantum states are.

$$S(A) \sim \mathrm{Area}(\partial B)$$ **short range RBM**

$$S(A) \sim \mathrm{Vol}(B)$$ **long range RBM**

Gives a measure of the long-range correlations in neural networks

[Deng, Li, Sarma, Phys. Rev. X, 7, 021021 (2017)]

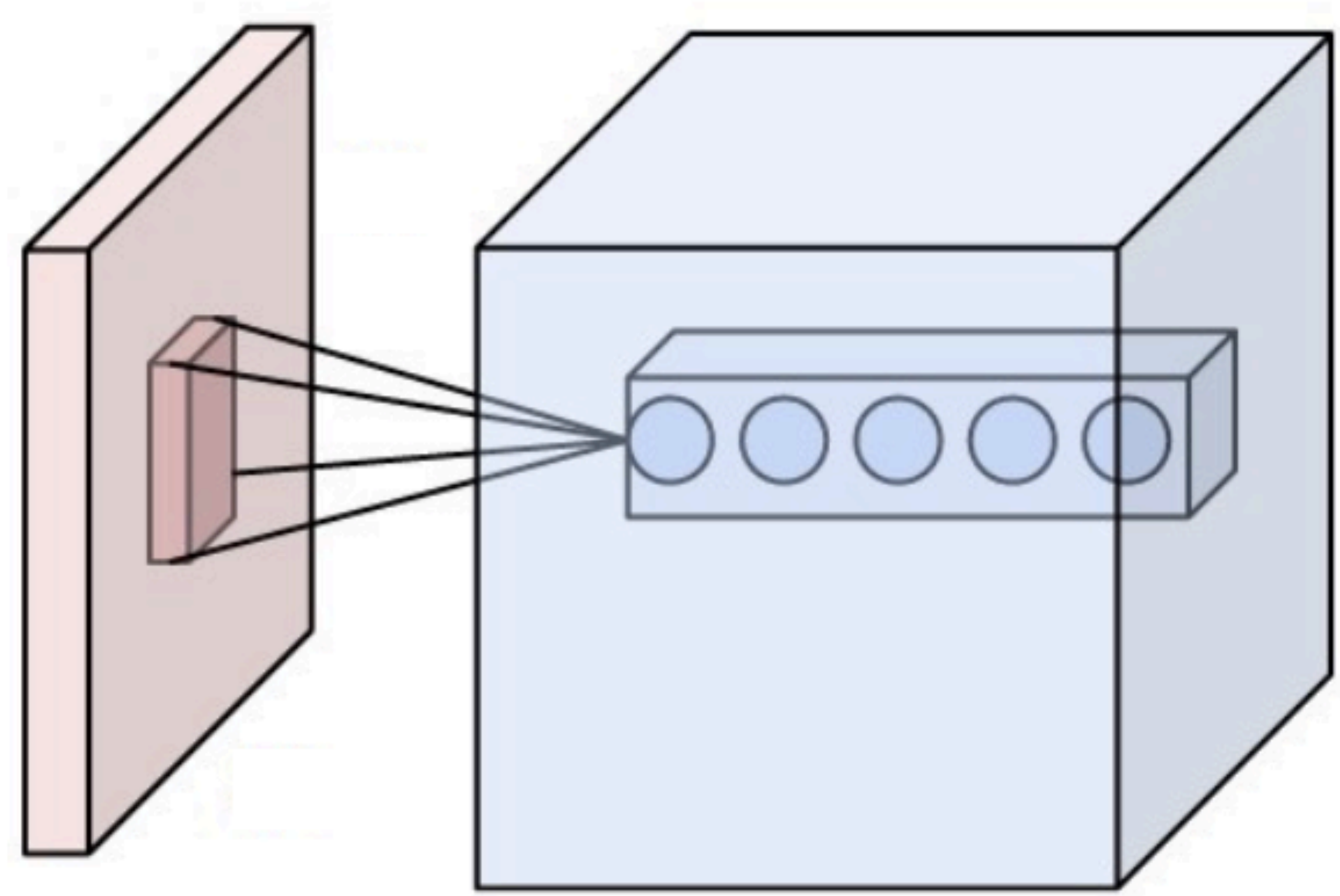
[Levine, Sharir, Cohen, Shashua, Phys. Rev. Lett, 122, 065301 (2019)]

Group equivariant CNNs and gauge theory

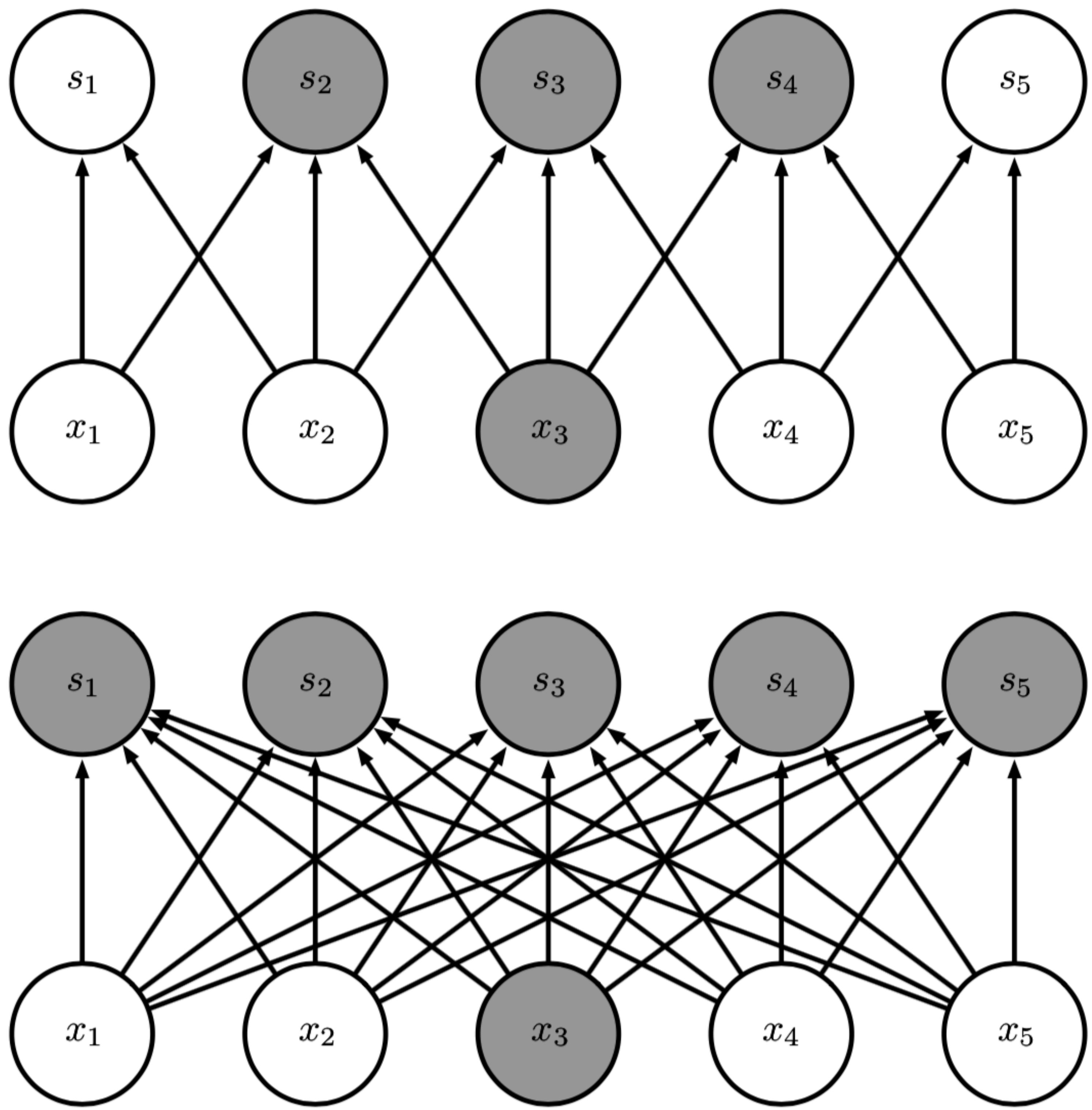
Convolutional Neural Networks

"Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers."

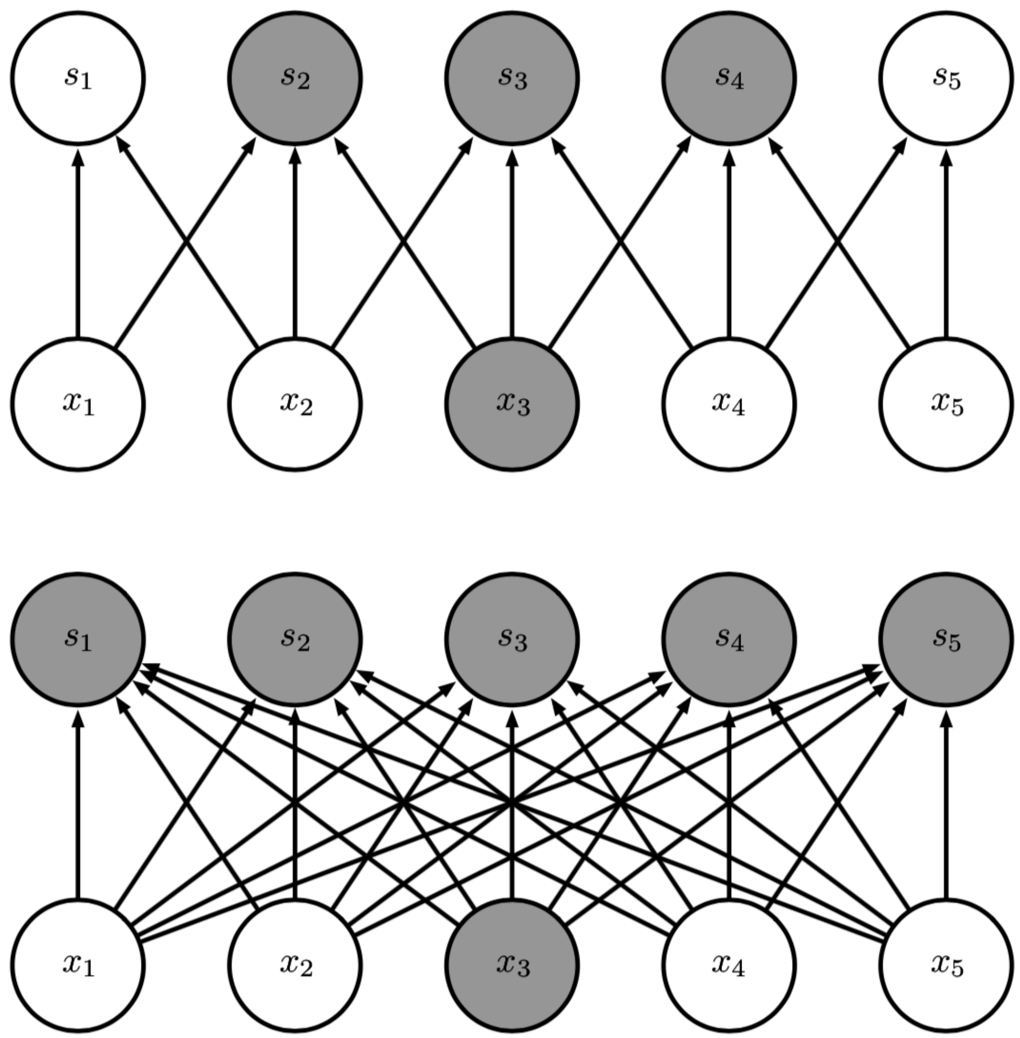
[Goodfellow, Bengio, Courville]



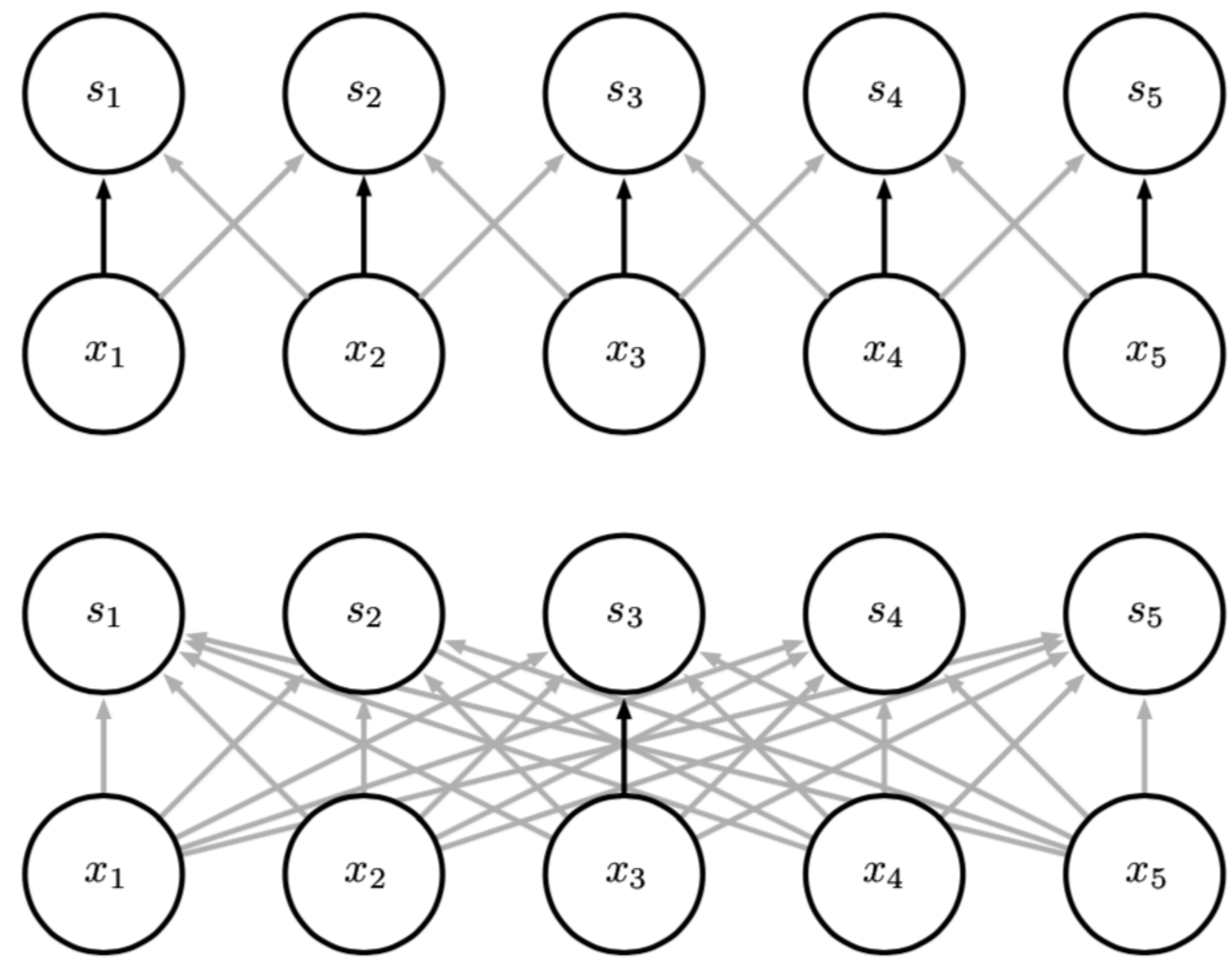
Sparse connectivity



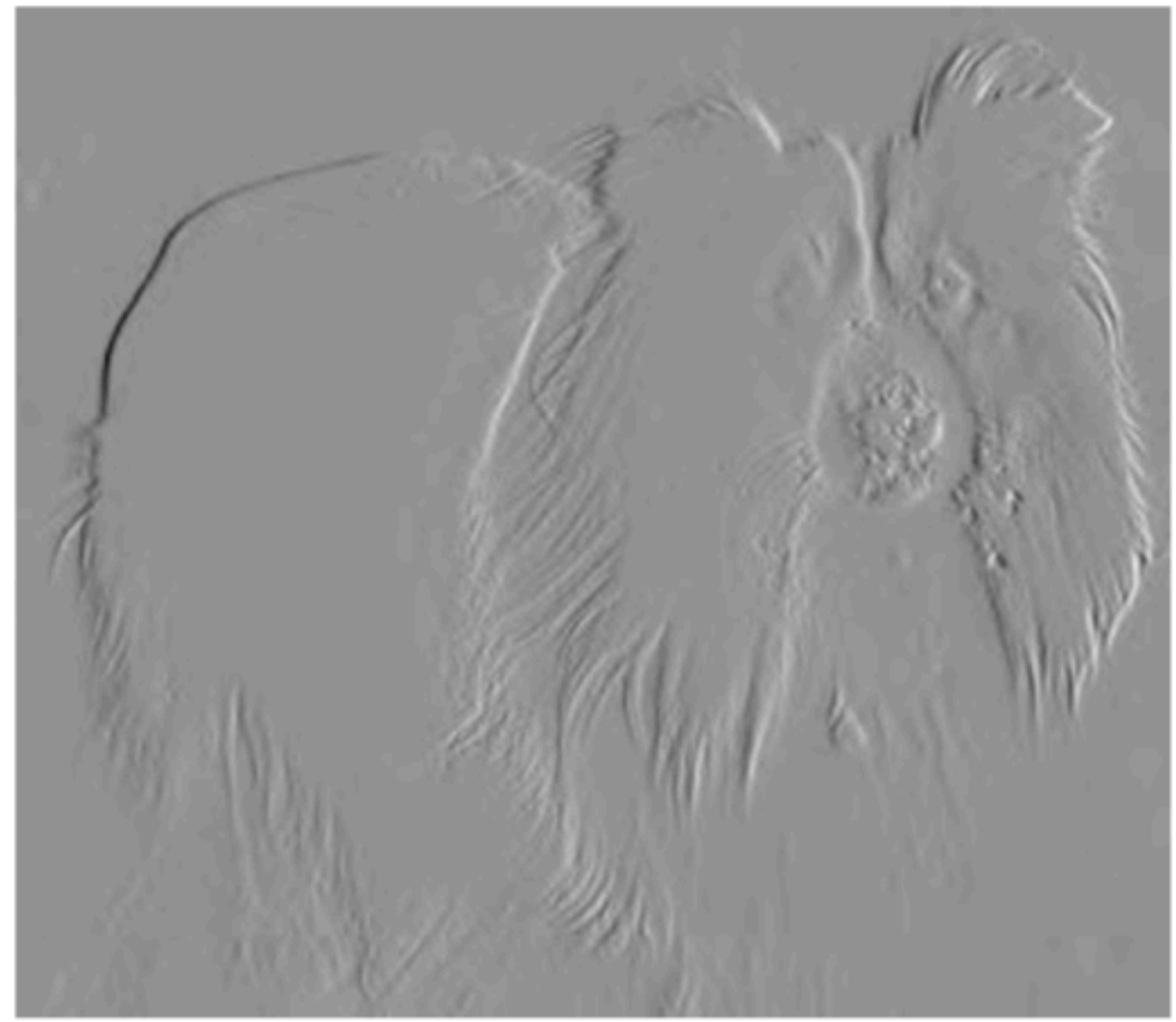
Sparse connectivity



Parameter sharing



Object detection using CNNs

Detection using only **vertically orinted edges**. Enormous efficiency improvement compared to matrix multiplication.

[Goodfellow, Bengio, Courville]

Mathematical structure

For each layer we have a **feature map**:

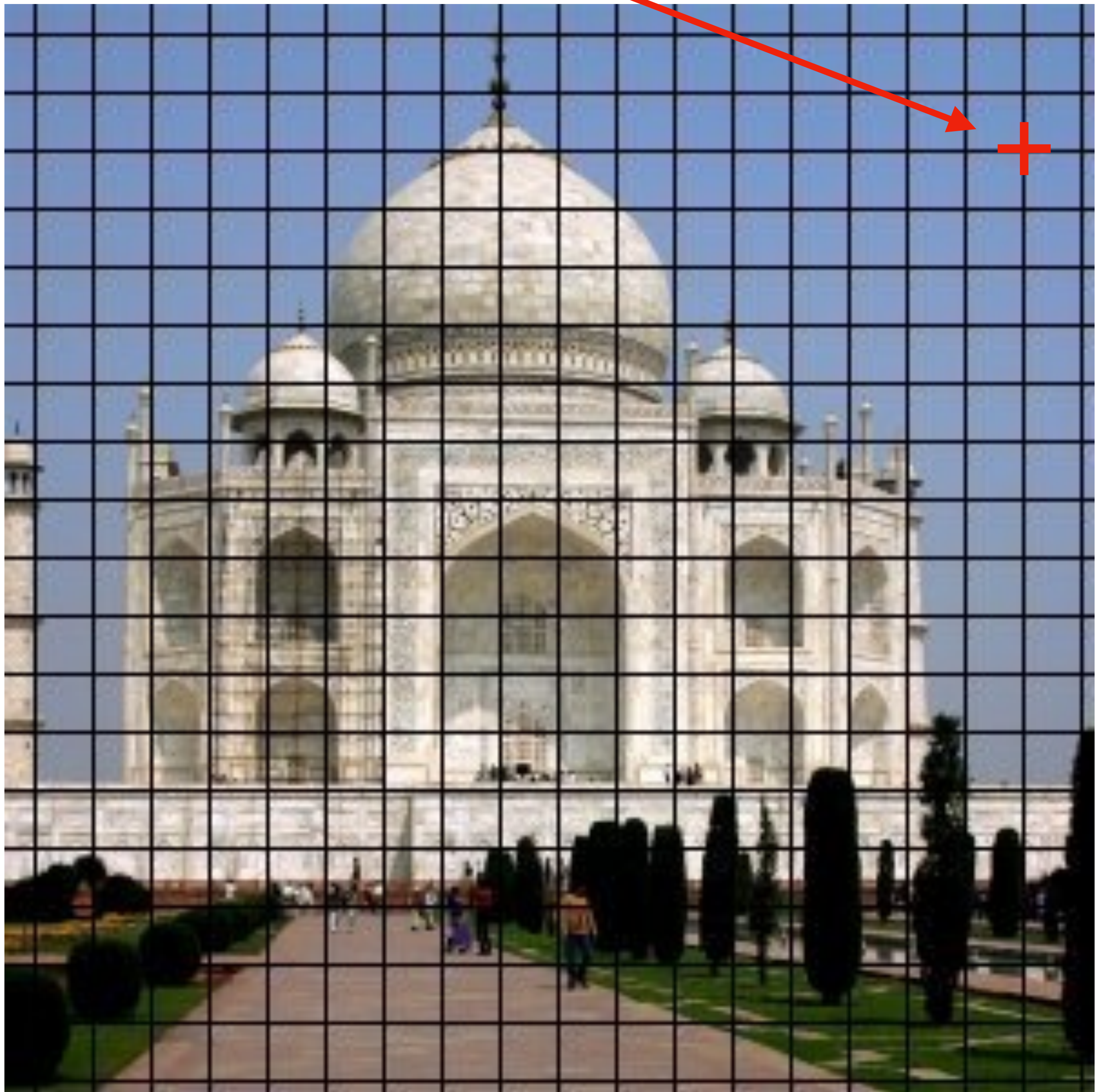
$$f : \mathbb{Z}^2 \to \mathbb{R}^K$$

Mathematical structure

For each layer we have a **feature map**:

no. of channels

$$f : \mathbb{Z}^2 \to \mathbb{R}^{K}$$



(p, q)

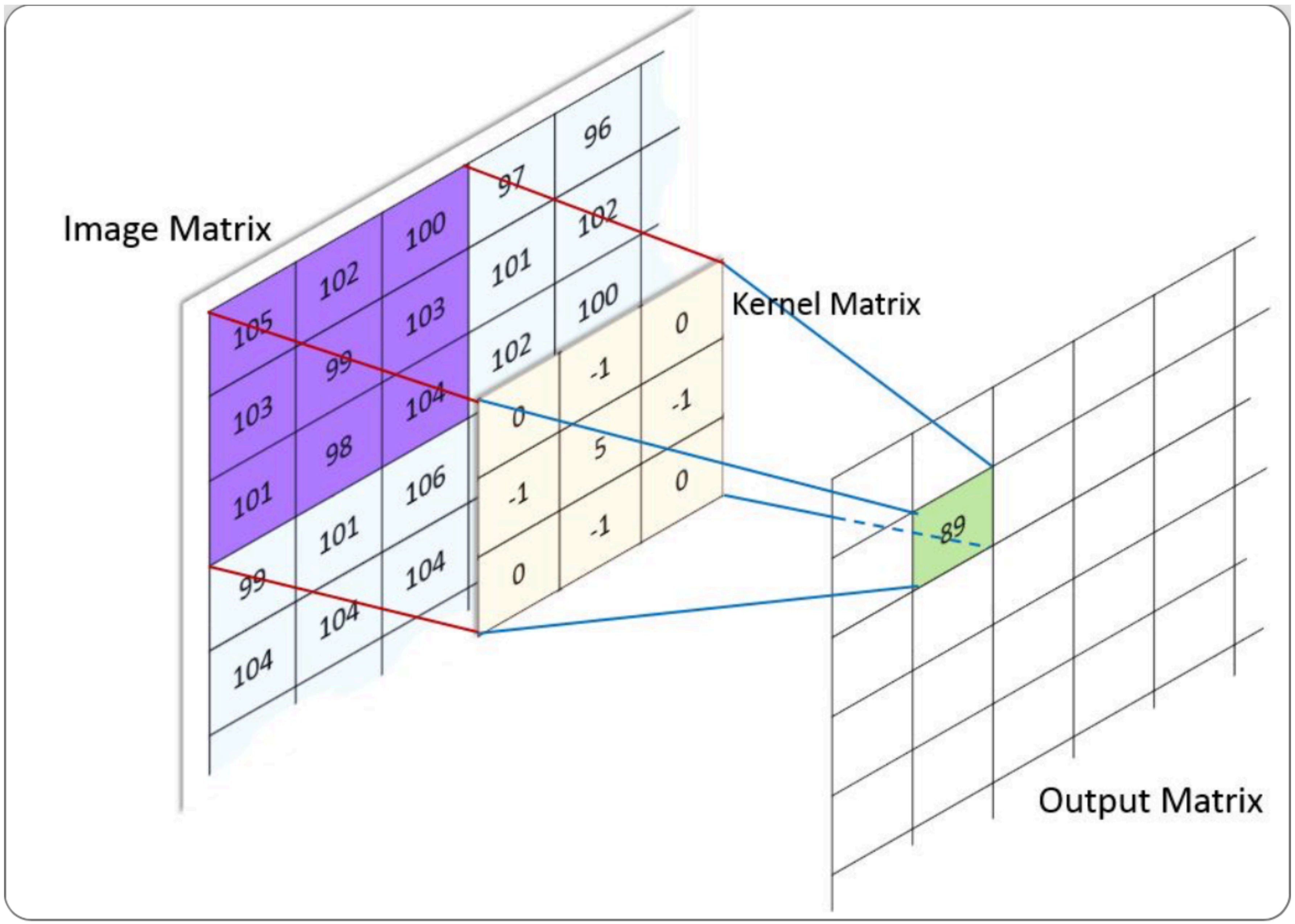
pixel coordinate

Kernel (filter): $\psi : \mathbb{Z}^2 \to \mathbb{R}^K$

Convolution: $$[f * \psi](x) = \sum_{y \in \mathbb{Z}^2} \sum_{k=1}^{K} f_k(y) \psi_k(x - y)$$

Kernel (filter): $\psi : \mathbb{Z}^2 \to \mathbb{R}^K$

Convolution: $$[f * \psi](x) = \sum_{y \in \mathbb{Z}^2} \sum_{k=1}^{K} f_k(y)\psi_k(x - y)$$



[Figure from machinelearninguru.com]

Kernel (filter): $\psi : \mathbb{Z}^2 \to \mathbb{R}^K$

Convolution: $$[f * \psi](x) = \sum_{y \in \mathbb{Z}^2} \sum_{k=1}^{K} f_k(y) \psi_k(x - y)$$

Translation map: $$[T(t)f](x) = f(x + t)$$

Kernel (filter): $\psi : \mathbb{Z}^2 \to \mathbb{R}^K$

Convolution: $$[f * \psi](x) = \sum_{y \in \mathbb{Z}^2} \sum_{k=1}^{K} f_k(y) \psi_k(x - y)$$

Translation map: $$[T(t)f](x) = f(x + t)$$

Convolution is equivariant

$$[T(t)f] * \psi = T(t)[f * \psi]$$

Note however, that convolution is **not equivariant with respect to other transformations** (e.g. rotations)

What about more general transformations?

Deep Learning for Non-Euclidean data [LeCun et al]

Classification of protein structures

Spherical signals - omnidirectional vision

Molecular geometries

Mathematical framework for neural networks

Weather and climate data

What about more general transformations?

Deep Learning for Non-Euclidean data

Classification of protein structures

Spherical signals - omnidirectional vision

Molecular geometries

Mathematical framework for neural networks

Weather and climate data

Strong motivation for developing group equivariant neural networks!

Group equivariant convolutions

Observation: $\mathbb{Z}^2$ is a **free abelian group** (w.r.t addition)

Replace $\mathbb{Z}^2$ by a general group G .

Feature map: $f : G \to \mathbb{R}^K$

Group equivariant convolutions

Observation: $\mathbb{Z}^2$ is a **free abelian group** (w.r.t addition)

Replace $\mathbb{Z}^2$ by a general group G .

Feature map: $f : G \rightarrow \mathbb{R}^K$

Convolution is now defined as

$$[f * \psi](g) = \sum_{h \in G} \sum_{k=1}^{K} f_k(h)\psi_k(gh) \qquad G \text{ discrete}$$

$$[f * \psi](g) = \int_G \sum_{k=1}^{K} f_k(h)\psi_k(gh)dh \qquad G \text{ continuous}$$

This is now G **-equivariant**

$$T(g)f * \psi = T(g)[f * \psi]$$

This is now G **-equivariant**

$$T(g)f * \psi = T(g)[f * \psi]$$

Definition: *A group equivariant CNN is a feed-forward neural network in which at each layer the linear map implements a* G *-equivariant convolution.*

[Kondor, Trivedi][Cohen, Welling]

General framework

G a group. $H \subset G$ subgroup.

Coset space: G/H

Vector space: $V \cong \mathbb{R}^n$

Representation: $\rho : H \to GL(V)$

General framework

G a group. $H \subset G$ subgroup.

Coset space: G/H

Vector space: $V \cong \mathbb{R}^n$

Representation: $\rho : H \to GL(V)$

Now consider: $P = (G \times V)/H$

This is an **equivalence class** with respect to

$$(g, v) \sim (gh, \rho(h^{-1})v)$$

This is a vector bundle:

$$\begin{tikzcd} V \arrow[r] & P \arrow[d, "p"] \\ & G/H \end{tikzcd}$$

Locally, it takes the form: $G/H \times V$

Sections of P **are maps:** $s : G/H \to P$ $\quad (p \circ s = \mathrm{Id})$

Locally, we can think of these as functions $f : G/H \to V$

Locally, we can think of these as functions $f : G/H \to V$

Example: $G = \mathbb{Z}^2$ $H = \{1\}$ $V = \mathbb{R}^K$

Feature maps are sections! $f : \mathbb{Z}^2 \to \mathbb{R}^K$

Locally, we can think of these as functions $f : G/H \to V$

Example: $G = \mathbb{Z}^2$ $H = \{1\}$ $V = \mathbb{R}^K$

Feature maps are sections! $f : \mathbb{Z}^2 \to \mathbb{R}^K$

General structure of group equivariant CNNs:

$$\left\{ \begin{array}{c} \text{Feature space of a} \\ G-\text{equivariant CNN} \end{array} \right\} \cong \left\{ \begin{array}{c} \text{Sections of vector} \\ \text{bundles } \; P \to G/H \end{array} \right\}$$

[Cohen, Geiger, Weiler]

$$\left\{ \begin{array}{c} \text{Feature space of a} \\ G-\text{equivariant CNN} \end{array} \right\} \cong \left\{ \begin{array}{c} \text{Sections of vector} \\ \text{bundles } P \to G/H \end{array} \right\}$$

Sections of $P \to G/H$ belong to the **induced representation**:

$$\mathcal{F} = \text{Ind}_H^G \rho = \{f : G \to V \mid f(gh) = \rho(h^{-1}) f(g)\}$$

$$\left\{ \begin{array}{c} \text{Feature space of a} \\ G-\text{equivariant CNN} \end{array} \right\} \quad \cong \quad \left\{ \begin{array}{c} \text{Sections of vector} \\ \text{bundles } \; P \rightarrow G/H \end{array} \right\}$$

Sections of $P \rightarrow G/H$ belong to the **induced representation**:

$$\mathcal{F} = \ \mathrm{Ind}_H^G \rho = \{ f : G \rightarrow V \mid f(gh) = \rho(h^{-1}) f(g) \}$$

$$\left\{ \begin{array}{c} \text{Maps between layers} \\ \text{in the CNN} \end{array} \right\} \cong \left\{ \begin{array}{c} G-\text{equivariant linear maps} \\ \text{between feature spaces } \mathcal{F} \rightarrow \mathcal{F}' \end{array} \right\}$$

$$\left\{ \begin{array}{c} \text{Feature space of a} \\ G-\text{equivariant CNN} \end{array} \right\} \cong \left\{ \begin{array}{c} \text{Sections of vector} \\ \text{bundles } P \to G/H \end{array} \right\}$$

Sections of $P \to G/H$ belong to the **induced representation**:

$$\mathcal{F} = \operatorname{Ind}_H^G \rho = \{f : G \to V \mid f(gh) = \rho(h^{-1}) f(g)\}$$

$$\left\{ \begin{array}{c} \text{Maps between layers} \\ \text{in the CNN} \end{array} \right\} \cong \left\{ \begin{array}{c} G-\text{equivariant linear maps} \\ \text{between feature spaces } \mathcal{F} \to \mathcal{F}' \end{array} \right\}$$

$$\mathrm{Hom}_G(\mathcal{F}, \mathcal{F}')$$

(intertwining operators)

Example: Spherical signals

$$G = SO(3)$$

$$H = SO(2)$$

$$G/H \cong S^2$$

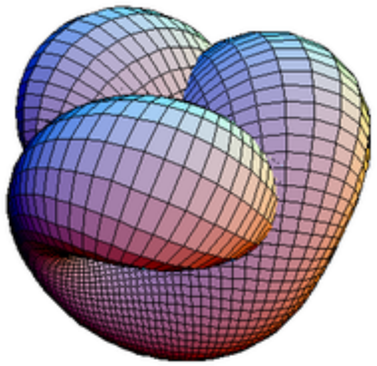
Feature maps

$$f : S^2 \to \mathbb{R}^K$$

Relevant for :

- Omnidirectional vision
- Weather and climate data (see e.g. [Cohen, Weiler, Kikanaoglu, Welling])
- Cosmology & astrophysics

Deep learning on manifolds & gauge theory



[Cohen, Weiler, Kikanaoglu, Welling][Cheng et al]

CNNs on arbitrary manifolds
require local equivariance

covariance w. r. t.
gauge transformations
(general coordinate transformations)

gauge equivariant
feature maps

Fields
Sections of vector bundles
(frame bundles)

“elementary feature types”
?

irreducible representations of G
elementary particles
(scalars, vectors, spinors…)

Are these the seeds of a deeper relation between neural networks and gauge theory?

Outlook

The **cross-fertilization** between deep learning, theoretical physics and mathematics is an exciting rapidly developing area of research

Outlook

The **cross-fertilization** between deep learning, theoretical physics and mathematics is an exciting rapidly developing area of research

- Is renormalization a universal principle for deep learning? (Callan-Symanzik equation?)
- Relation with Quantum Information Theory?
- Can we realize a neural network as a (quantum) dynamical system?
- Relation with optimal transport theory and information geometry?
- Can we implement symmetries and conservation laws?
- A spacetime perspective of Deep Neural Networks?
- Emergent phenomena?

Exciting times ahead!
Thank you!