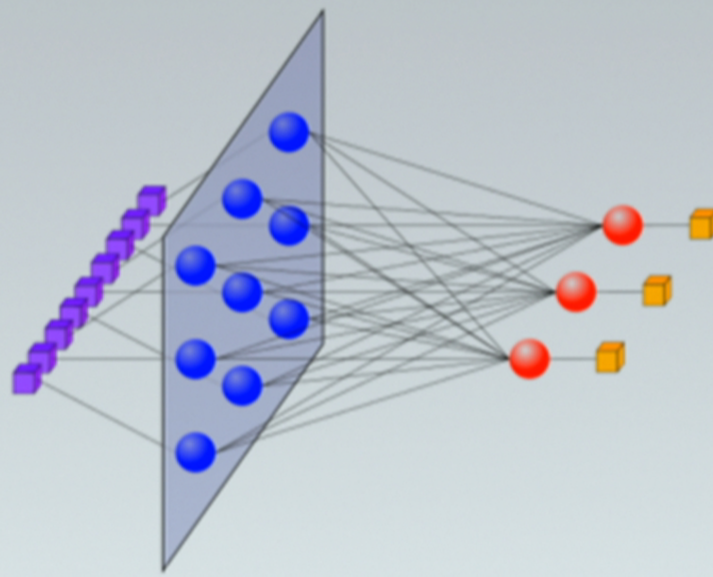
Title: Learning Thermodynamics with Boltzmann Machines

Date: Aug 11, 2016 02:45 PM

URL: http://pirsa.org/16080017

Abstract: The introduction of neural networks with deep architecture has led to a revolution, giving rise to a new wave of technologies empowering our modern society. Although data science has been the main focus, the idea of generic algorithms which automatically extract features and representations from raw data is quite general and applicable in multiple scenarios. Motivated by the effectiveness of deep learning algorithms in revealing complex patterns and structures underlying data, we are interested in exploiting such tool in the context of many-body physics. I will first introduce the Boltzmann Machine, a stochastic neural network that has been extensively used in the layers of deep architectures. I will describe how such network can be used for modelling thermodynamic observables for physical systems in thermal equilibrium, and show that it can faithfully reproduce observables for the 2 dimensional Ising model. Finally, I will discuss how to adapt the same network for implementing the classical computation required to perform quantum error correction in the 2D toric code.

Learning Thermodynamics with Boltzmann Machines



Giacomo Torlai
University of Waterloo

University of Waterloo

Perimeter Institute
August 11th 2016

PERIMETER INSTITUTE

Outline

Boltzmann Machines

Learning Thermodynamics of Spin Systems

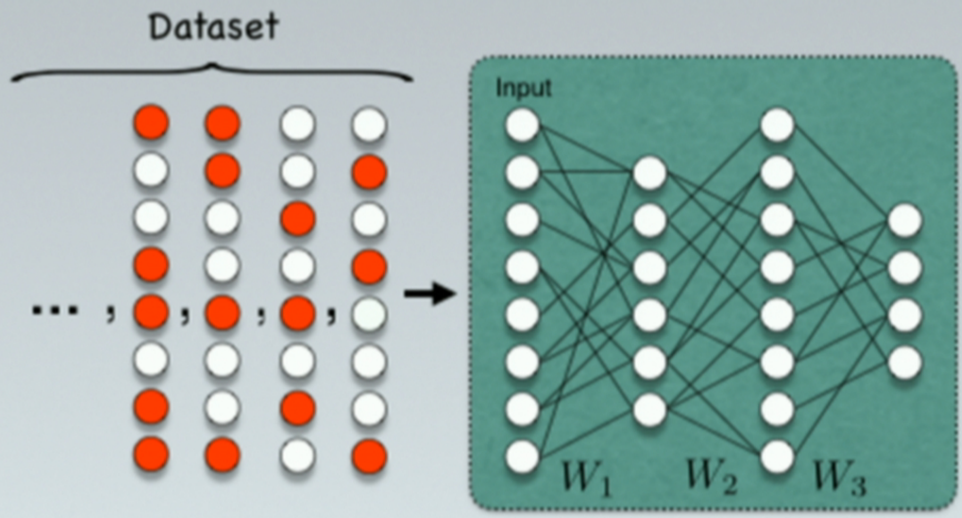
Machine Learning Decoder

Conclusions

gtorlai@uwaterloo.ca

Boltzmann Machines

General Framework



Network Parameters

$$\lambda = \{W_1, W_2, W_3\}$$

Given a dataset $\mathcal{D}$ of input vectors and a target probability distribution p underlying the data, the **goal** is to find the set of parameters $\widetilde{\lambda}$ which minimize/maximize a suitable objective function.

gtorlai@uwaterloo.ca

Boltzmann Machines

Discriminative Training

Dataset: $\mathbb{D} = \{(\mathbf{v}_k, \ell_k)\} \quad k = 1, \ldots, N$

Example: discriminate 0 from 1.

ℓ 0 1 1 0 0

$\mathbf{v}$ 

Target probability $p(\ell \,|\, \mathbf{v})$

Criterion: Maximize the probability of finding the correct label of an input.

gtorlai@uwaterloo.ca

Boltzmann Machines

Discriminative Training

Dataset: $\mathbb{D} = \{(\mathbf{v}_k, \ell_k)\} \quad k = 1, \ldots, N$

Example: discriminate 0 from 1.

ℓ	0	1	1	0	0

$\mathbf{v}$ 

Target probability $p(\ell \,|\, \mathbf{v})$

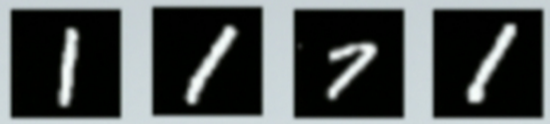
Criterion: Maximize the probability of finding the correct label of an input.

Generative Training

Dataset: $\mathbb{D} = \{\mathbf{v}_k\} \quad k = 1, \ldots, N$

Example: learn the digit 1.

ℓ No Label

$\mathbf{v}$ 

Target probability $p(\mathbf{v})$

Criterion: Maximize the probability that the network would generate the input.

gtorlai@uwaterloo.ca

Boltzmann Machines

Discriminative Training

Dataset: $\mathbb{D} = \{(\mathbf{v}_k, \ell_k)\} \quad k = 1, \dots, N$

Example: discriminate 0 from 1.

ℓ 0 1 1 0 0

$\mathbf{v}$

Target probability $p(\ell \,|\, \mathbf{v})$

Criterion: Maximize the probability of finding the correct label of an input.

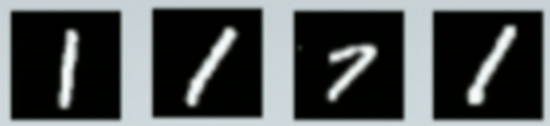
Generative Training

Dataset: $\mathbb{D} = \{\mathbf{v}_k\} \quad k = 1, \dots, N$

Example: learn the digit 1.

ℓ No Label

$\mathbf{v}$



Target probability $p(\mathbf{v})$

Criterion: Maximize the probability that the network would generate the input.

gtorlai@uwaterloo.ca

Boltzmann Machines

Restricted Boltzmann Machine

$$H_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}, \mathbf{h}) = -\mathbf{h}^\top \mathbf{W} \boldsymbol{\sigma} - \mathbf{b}^\top \boldsymbol{\sigma} - \mathbf{c}^\top \mathbf{h}$$

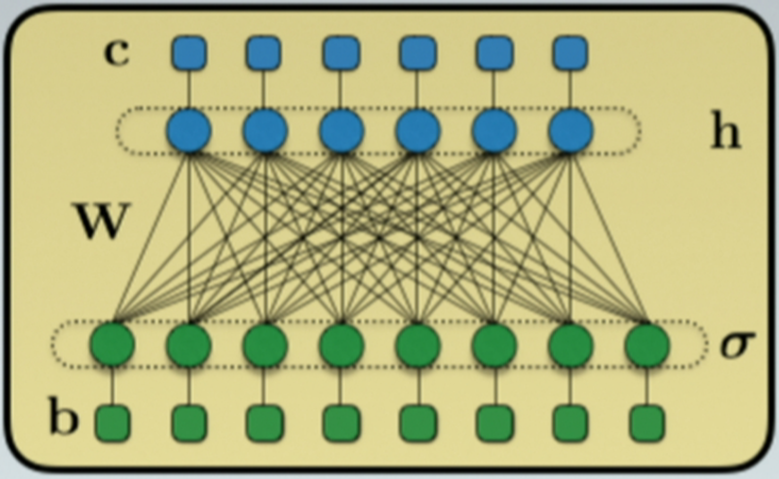
$$p_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}, \mathbf{h}) = \frac{1}{Z_{\boldsymbol{\lambda}}} \mathrm{e}^{-H_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}, \mathbf{h})}$$

$$p_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}) = \sum_{\boldsymbol{h}} p_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}, \boldsymbol{h}) = \frac{1}{Z_{\boldsymbol{\lambda}}} \mathrm{e}^{-\mathcal{E}_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})}$$

$$p_{\boldsymbol{\lambda}}(\boldsymbol{\sigma} \,|\, \boldsymbol{h}) = \prod_j p_{\boldsymbol{\lambda}}(\sigma_j \,|\, \boldsymbol{h})$$

$$p_{\boldsymbol{\lambda}}(\boldsymbol{h} \,|\, \boldsymbol{\sigma}) = \prod_i p_{\boldsymbol{\lambda}}(h_i \,|\, \boldsymbol{\sigma})$$

$\lambda = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$



gtorlai@uwaterloo.ca

Boltzmann Machines

Restricted Boltzmann Machine

$$H_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}, \mathbf{h}) = -\mathbf{h}^\top \mathbf{W} \boldsymbol{\sigma} - \mathbf{b}^\top \boldsymbol{\sigma} - \mathbf{c}^\top \mathbf{h}$$

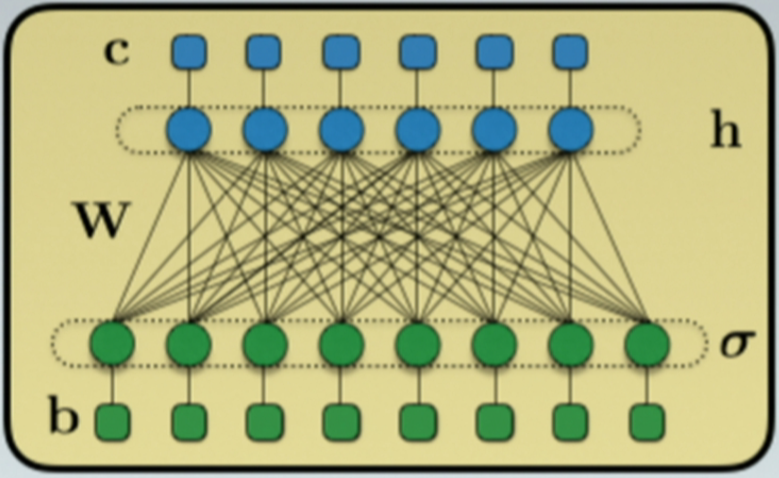
$$p_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}, \mathbf{h}) = \frac{1}{Z_{\boldsymbol{\lambda}}} \mathrm{e}^{-H_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}, \mathbf{h})}$$

$$p_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}) = \sum_{\boldsymbol{h}} p_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}, \boldsymbol{h}) = \frac{1}{Z_{\boldsymbol{\lambda}}} \mathrm{e}^{-\mathcal{E}_{\boldsymbol{\lambda}}(\boldsymbol{\sigma})}$$

$$p_{\boldsymbol{\lambda}}(\boldsymbol{\sigma} \,|\, \boldsymbol{h}) = \prod_j p_{\boldsymbol{\lambda}}(\sigma_j \,|\, \boldsymbol{h})$$

$$p_{\boldsymbol{\lambda}}(\boldsymbol{h} \,|\, \boldsymbol{\sigma}) = \prod_i p_{\boldsymbol{\lambda}}(h_i \,|\, \boldsymbol{\sigma})$$

$$\lambda = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$$

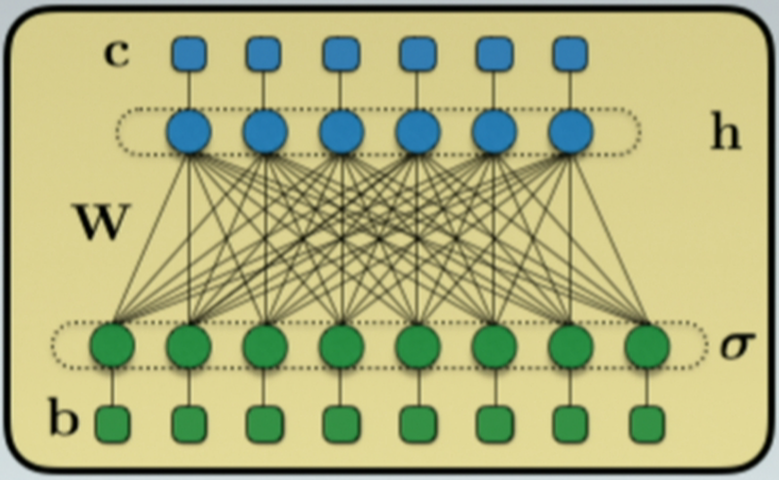


gtorlai@uwaterloo.ca

Boltzmann Machines

Training RBMs

$\lambda = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$



Objective function:

$$\mathbb{KL}(p_{\text{data}} \,||\, p_{\boldsymbol{\lambda}}) \sim -\frac{1}{|\mathcal{D}|}\sum_{k=1}^{|\mathcal{D}|} \log p_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}_k)$$

Goal
Find $\widetilde{\lambda}$ that minimizes the Kullback-Leibler divergence.

Stochastic Gradient Descent

$$\boldsymbol{\lambda}_j \leftarrow \boldsymbol{\lambda}_j - \frac{\eta}{|\mathcal{D}^{[b]}|}\sum_{\boldsymbol{\sigma}\in\mathcal{D}^{[b]}} \nabla_{\boldsymbol{\lambda}_j}\mathbb{KL}\left(p_{\text{data}} \,||\, p_{\boldsymbol{\lambda}}\right)$$

$$\nabla_{\boldsymbol{W}}\mathbb{KL}\left(p_{\text{data}} \,||\, p_{\boldsymbol{\lambda}}\right) = -\langle \boldsymbol{\sigma}\,\boldsymbol{h}^{\top}\rangle_{p_{\boldsymbol{\lambda}}(\boldsymbol{h}\,|\,\boldsymbol{\sigma})} + \langle \boldsymbol{\sigma}\,\boldsymbol{h}^{\top}\rangle_{p_{\boldsymbol{\lambda}}(\boldsymbol{\sigma},\boldsymbol{h})}$$

Trivial — Contrastive Divergence

G. E. Hinton, Neural Computation **14**, 1771 (2002)

gtorlai@uwaterloo.ca

Learning Thermodynamics of Spin Systems

2d Ising Model

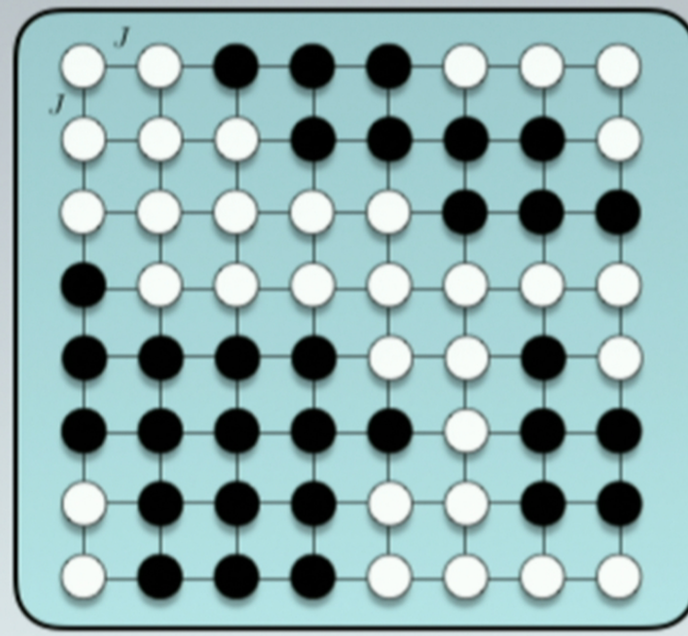
$$H_I = -J \sum_{\langle ij \rangle} \sigma_i \sigma_j$$

$$p_{\mathrm{target}}(\boldsymbol{\sigma}; T) = \frac{1}{Z} \mathrm{e}^{-\frac{1}{T} H_I}$$

Generate the dataset probability distribution $p_{\mathrm{data}}(\mathbf{v})$ by sampling from $p_{\mathrm{target}}(\mathbf{v}, T)$ at different temperatures.

$$p_{\mathrm{data}}(\boldsymbol{\sigma}) = \frac{1}{|\mathcal{D}|} \sum_{k=1}^{|\mathcal{D}|} \delta_{\boldsymbol{\sigma}, \boldsymbol{\sigma}_k}$$

$$T = 1.0, \ldots, 2.269, \ldots, 3.54$$



gtorlai@uwaterloo.ca

Learning Thermodynamics of Spin Systems

2d Ising Model

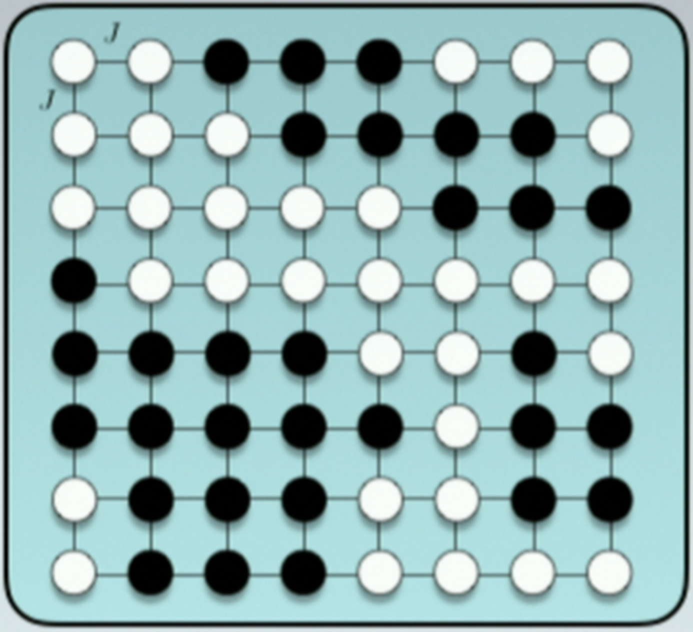
$$H_I = -J \sum_{\langle ij \rangle} \sigma_i \sigma_j$$

$$p_{\text{target}}(\boldsymbol{\sigma}; T) = \frac{1}{Z} \mathrm{e}^{-\frac{1}{T} H_I}$$

Generate the dataset probability distribution $p_{\text{data}}(\mathbf{v})$ by sampling from $p_{\text{target}}(\mathbf{v}, T)$ at different temperatures.

$$p_{\text{data}}(\boldsymbol{\sigma}) = \frac{1}{|\mathcal{D}|} \sum_{k=1}^{|\mathcal{D}|} \delta_{\boldsymbol{\sigma}, \boldsymbol{\sigma}_k}$$

$$T = 1.0, \ldots, 2.269, \ldots, 3.54$$

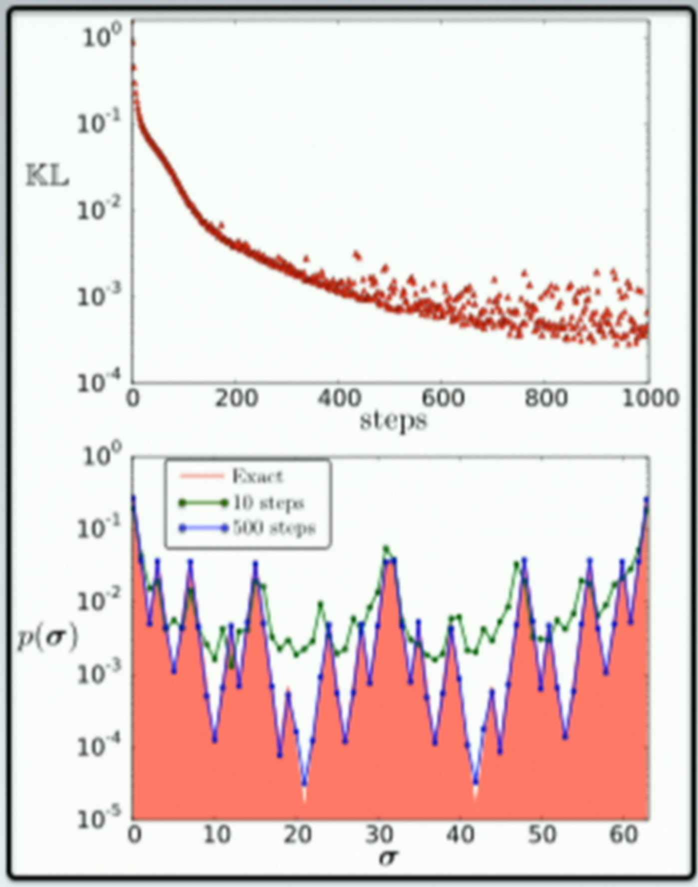


gtorlai@uwaterloo.ca

Learning Thermodynamics of Spin Systems

A toy model: Ising chain

GT and R. Melko, arXiv 1606:02718 (2016)



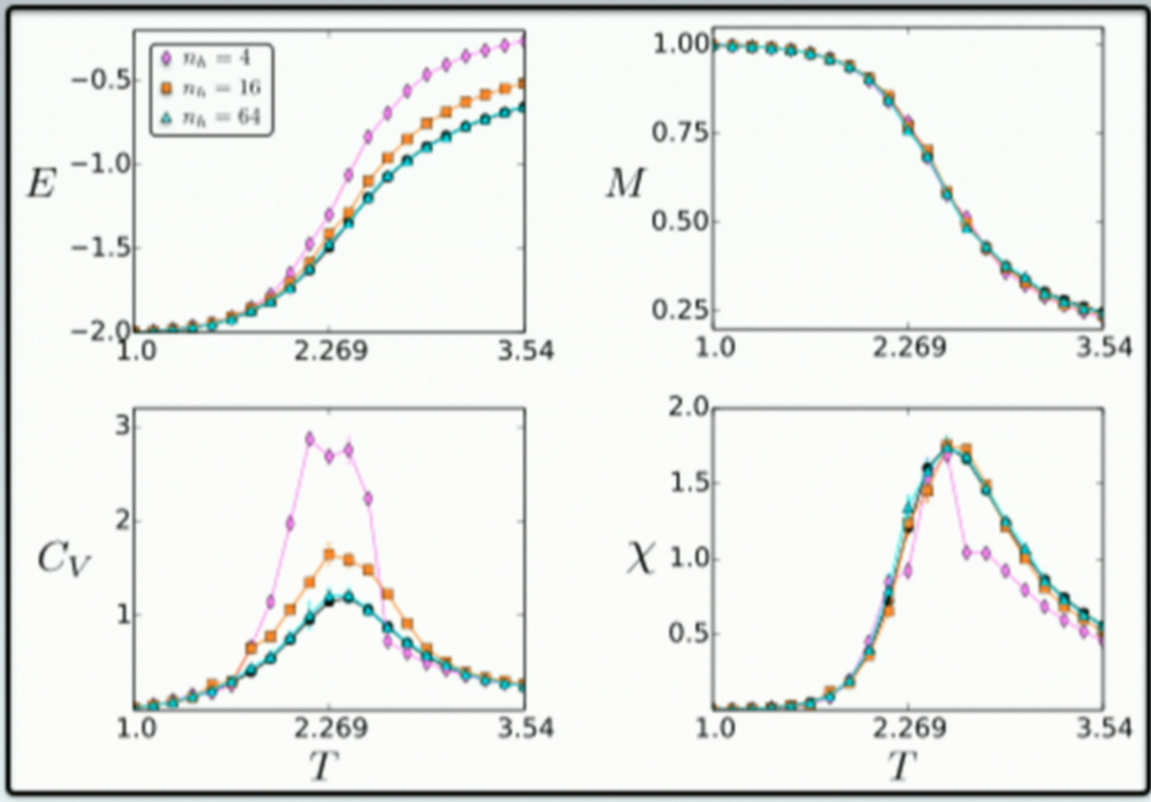
gtorlai@uwaterloo.ca

Learning Thermodynamics of Spin Systems

Thermodynamics of 2d Ising

GT and R. Melko, arXiv 1606:02718 (2016)

L = 8



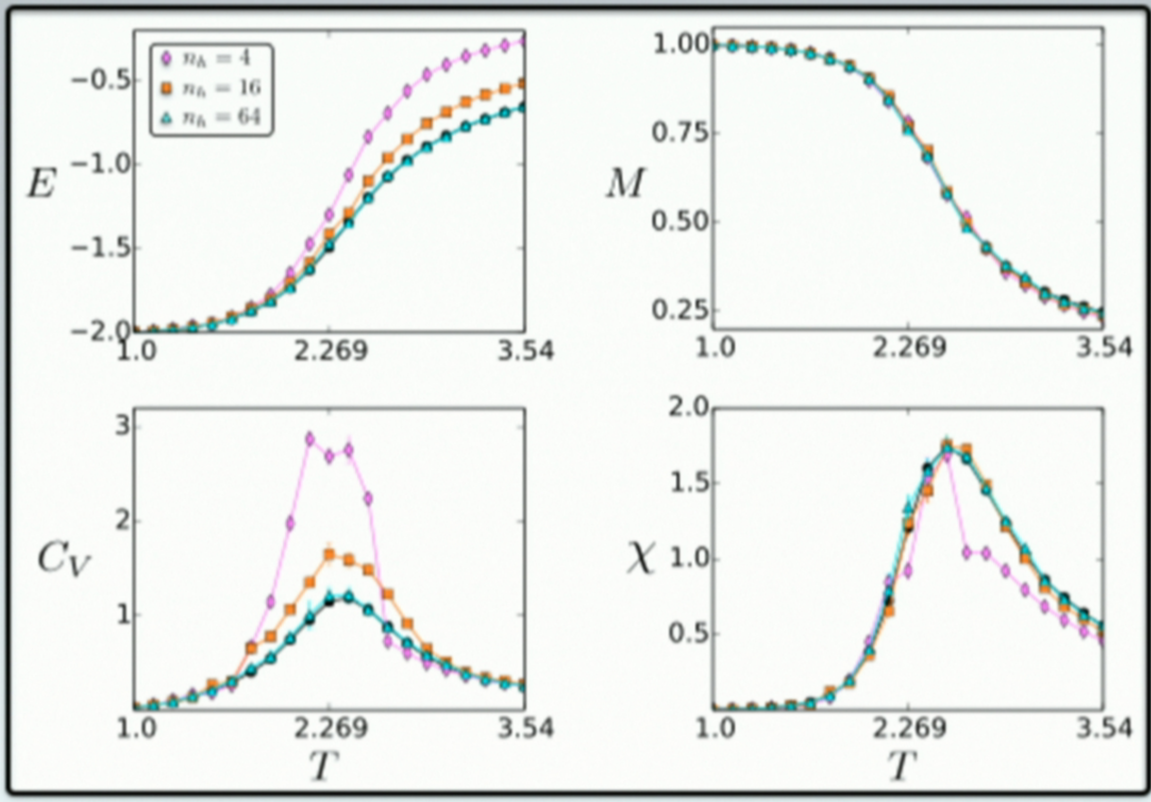
gtorlai@uwaterloo.ca

Learning Thermodynamics of Spin Systems

Thermodynamics of 2d Ising

GT and R. Melko, arXiv 1606:02718 (2016)

L = 8

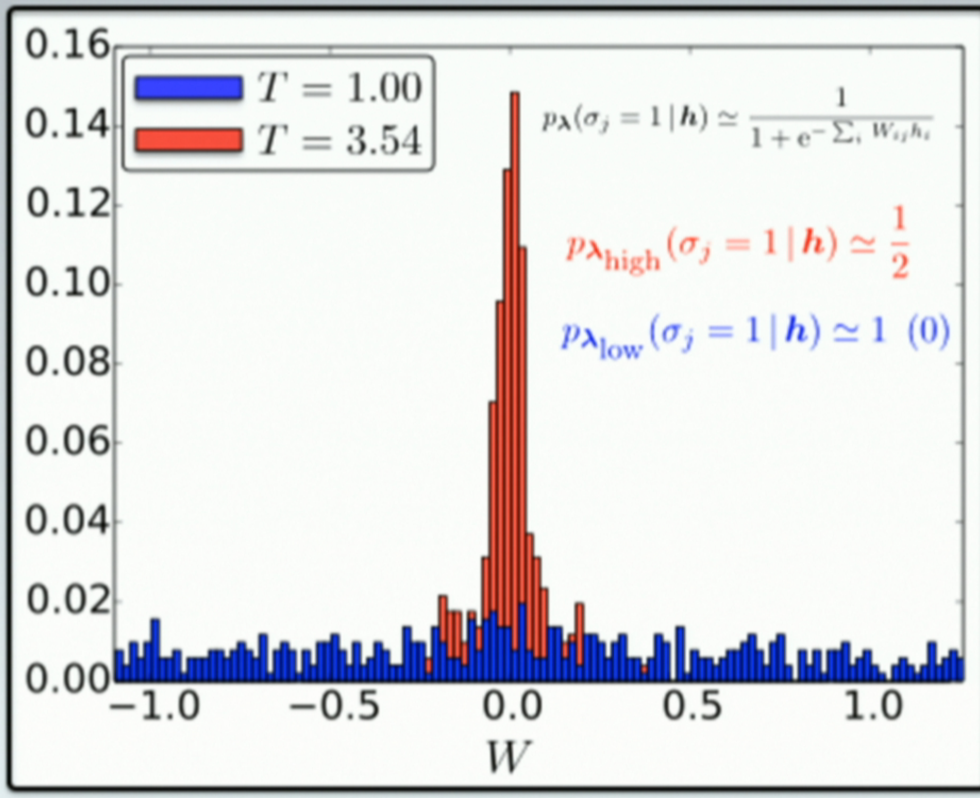


gtorlai@uwaterloo.ca

Learning Thermodynamics of Spin Systems

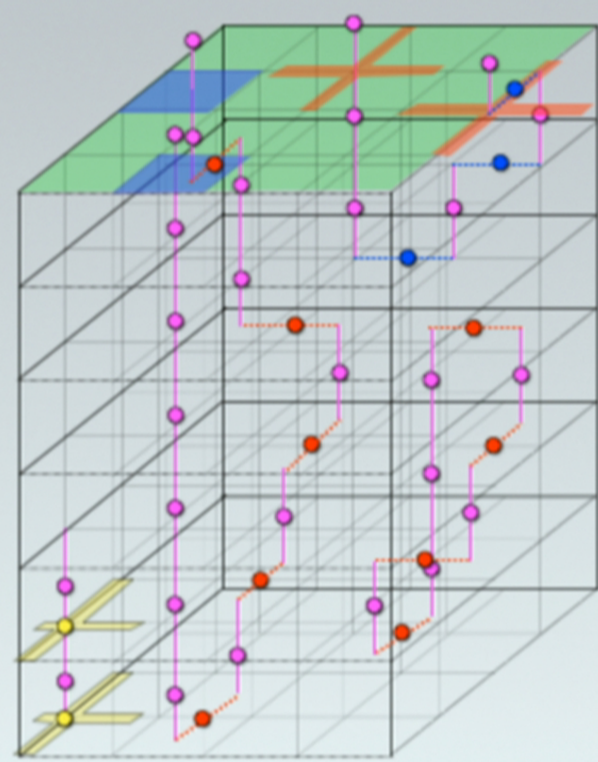
What is the hidden layer doing?

GT and R. Melko, arXiv 1606:02718 (2016)



gtorlai@uwaterloo.ca

Machine Learning Decoder

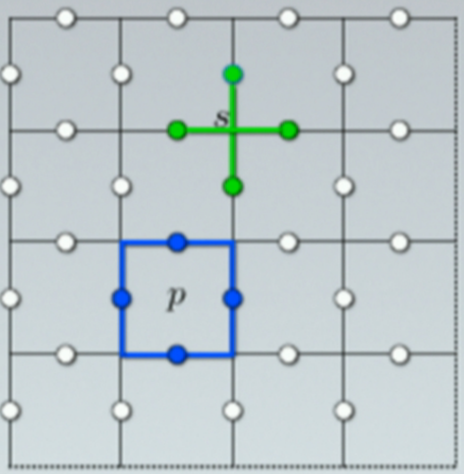


gtorlai@uwaterloo.ca

Machine Learning Decoder

The 2d Toric Code

N qubits are placed on the edges of a $L \times L$ square lattice embedded on a torus.



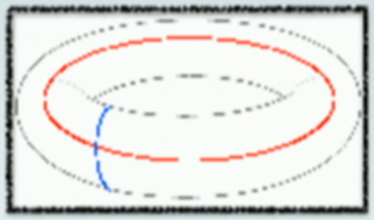
Introduce the stabilizers operators:

$$\hat{Z}_p = \bigotimes_{j \in \partial p} \hat{Z}_j \qquad \hat{X}_s = \bigotimes_{j \in s} \hat{X}_j \qquad [\hat{Z}_p, \hat{X}_s] = 0 \ \ \forall p, s$$

for each plaquette p and each vertex s .

$$H = -\sum_p \hat{Z}_p - \sum_s \hat{X}_s$$

- Highly degenerate ground state.
- Quantum information encoded in the topological sector.
- Logical operators are the non-trivial loops on the torus.



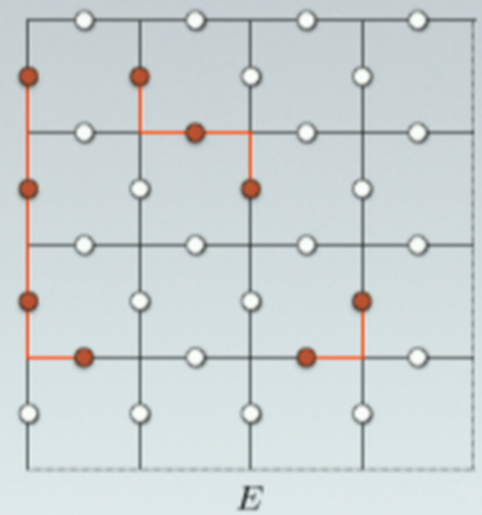
A. Yu. Kitaev, Annals of Physics 303 (2003) 2-30.

gtorlai@uwaterloo.ca

Machine Learning Decoder

Error Correction

Stabilizers operators detect endpoints of the error chains in the code (syndrome).



E Dennis *et al*, J. Math. Phys. 43, 4452 (2002)

gtorlai@uwaterloo.ca

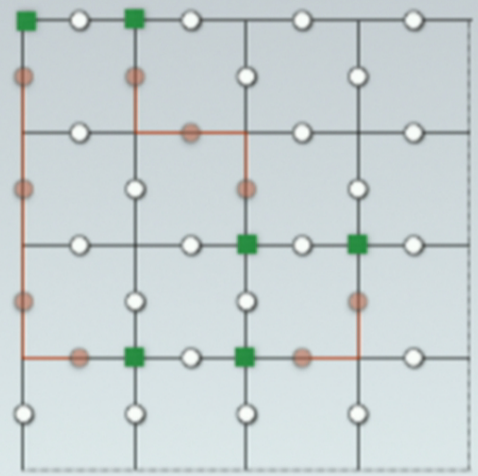
Machine Learning Decoder

Error Correction

Stabilizers operators detect endpoints of the error chains in the code (syndrome).

Algorithm

Extract the Syndrome



E

E Dennis *et al*, J. Math. Phys. 43, 4452 (2002)

gtorlai@uwaterloo.ca

Machine Learning Decoder

Error Correction

Stabilizers operators detect endpoints of the error chains in the code (syndrome).

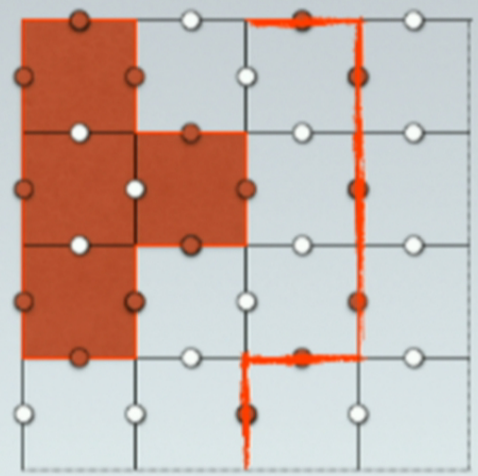
Algorithm

Extract the Syndrome

Generate Recovery Chain

Recover and Extract Cycle

Compute Homology Sector ⟶ FAILURE!



$E \oplus E'$

Given an initial error chain E and a recovery chain E', the error correction fails if the resulting operator $E \oplus E'$ acting on the code is a loop with non-trivial homology.

E Dennis *et al*, J. Math. Phys. 43, 4452 (2002)

gtorlai@uwaterloo.ca

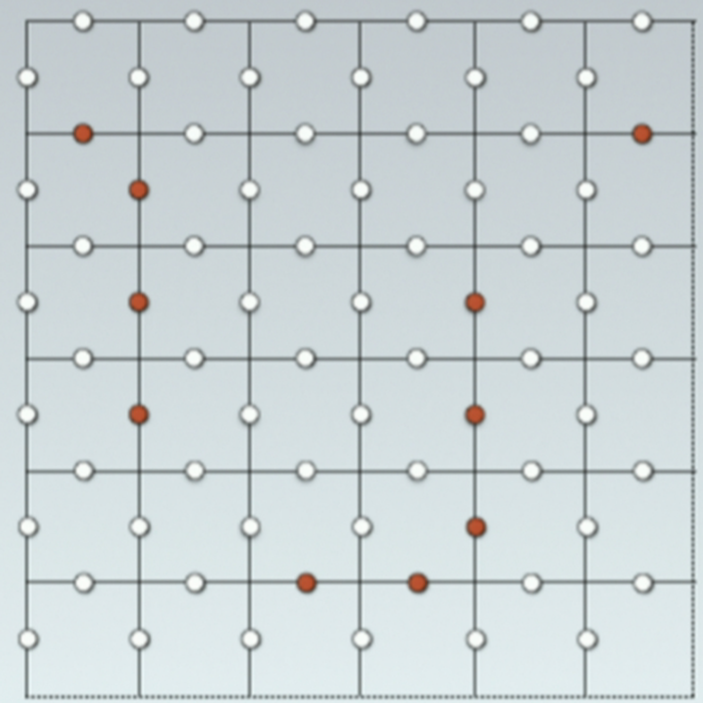
Machine Learning Decoder

Minimum-Weight Perfect Matching (and its failure)

MWPM connects the chain endpoints pairwise using the minimum number of links.

Low Error Regime **Corrected**

High Error Regime

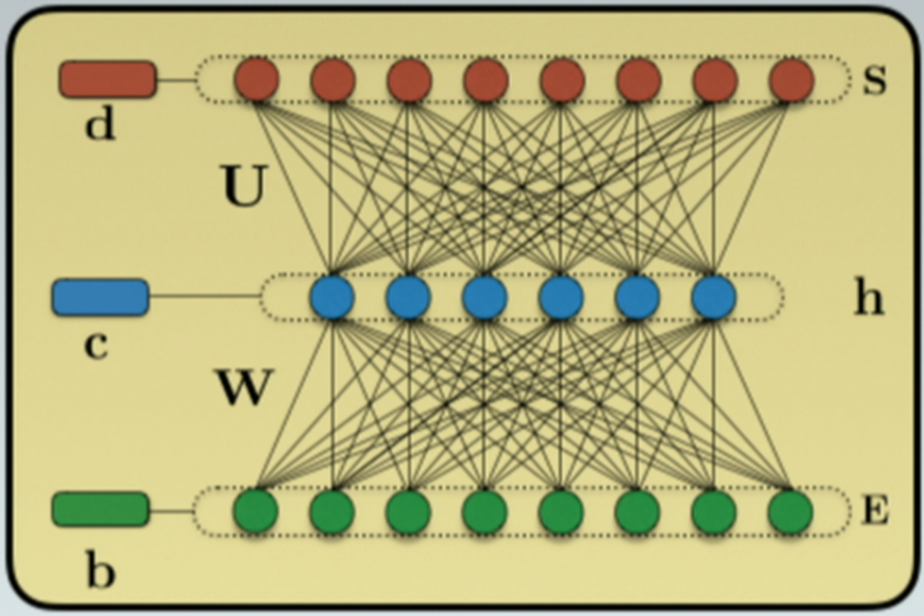


AG Fowler *et al*, PRA 86, 032324 (2012)

gtorlai@uwaterloo.ca

Machine Learning Decoder

Joint Restricted Boltzmann Machine



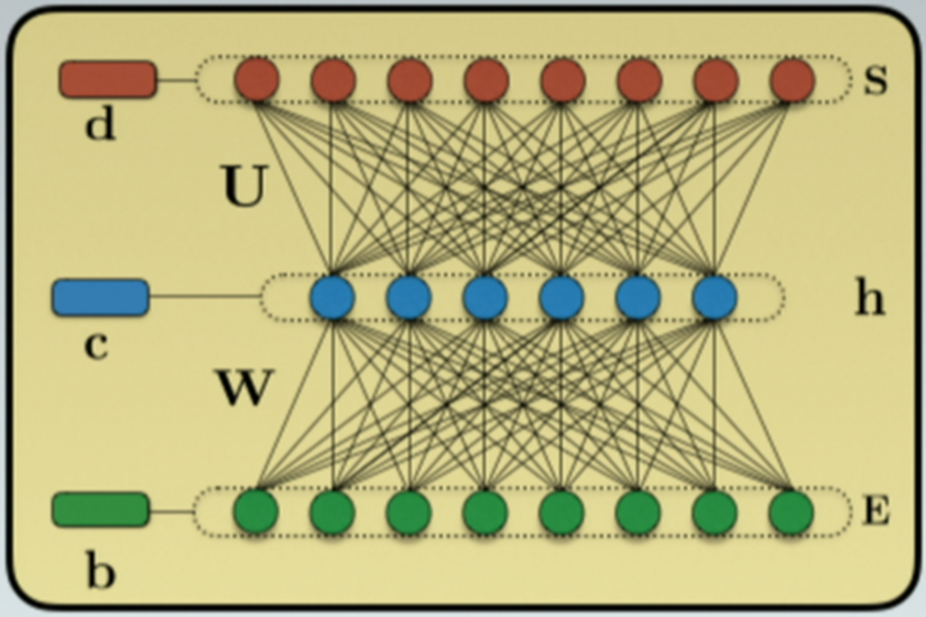
$$\lambda = \{\mathbf{W}, \mathbf{U}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$$

$$H_{\boldsymbol{\lambda}}(\mathbf{E}, \mathbf{h}, \mathbf{S}) = -\mathbf{h}^{\top}\mathbf{W}\,\mathbf{E} - \mathbf{h}^{\top}\mathbf{U}\,\mathbf{S} - \mathbf{b}^{\top}\mathbf{E} - \mathbf{c}^{\top}\mathbf{h} - \mathbf{d}^{\top}\mathbf{S}$$

gtorlai@uwaterloo.ca

Machine Learning Decoder

Joint Restricted Boltzmann Machine



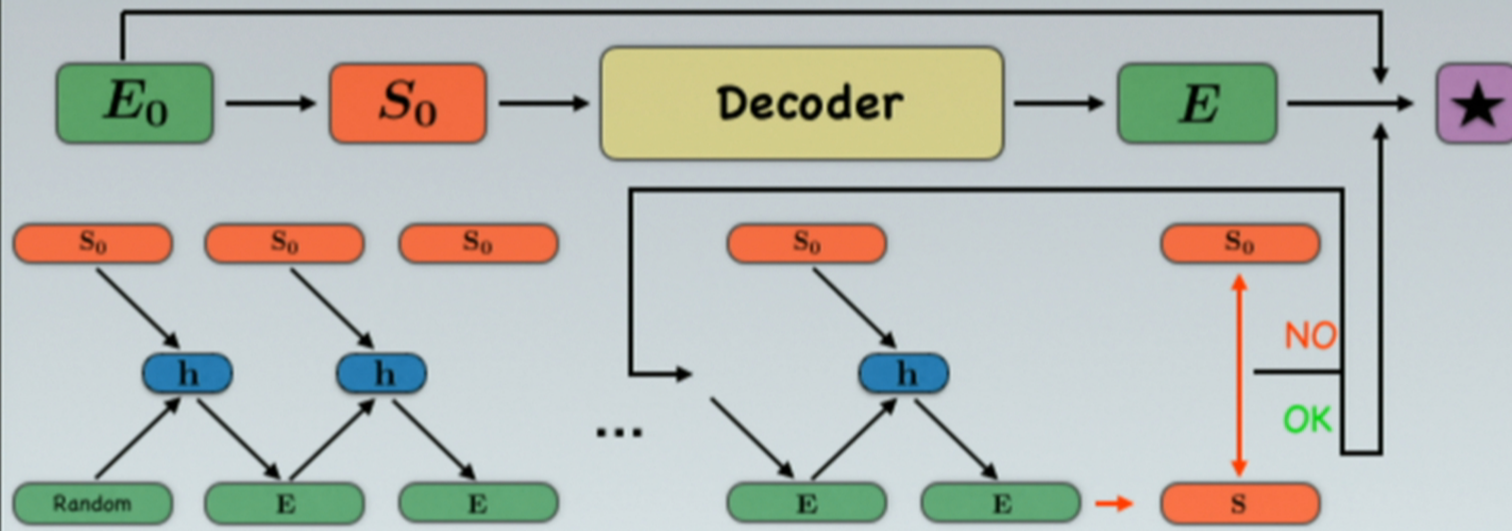
$$\lambda = \{\mathbf{W}, \mathbf{U}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$$

$$H_{\boldsymbol{\lambda}}(\mathbf{E}, \mathbf{h}, \mathbf{S}) = -\mathbf{h}^{\top}\mathbf{W}\,\mathbf{E} - \mathbf{h}^{\top}\mathbf{U}\,\mathbf{S} - \mathbf{b}^{\top}\mathbf{E} - \mathbf{c}^{\top}\mathbf{h} - \mathbf{d}^{\top}\mathbf{S}$$

gtorlai@uwaterloo.ca

Machine Learning Decoder

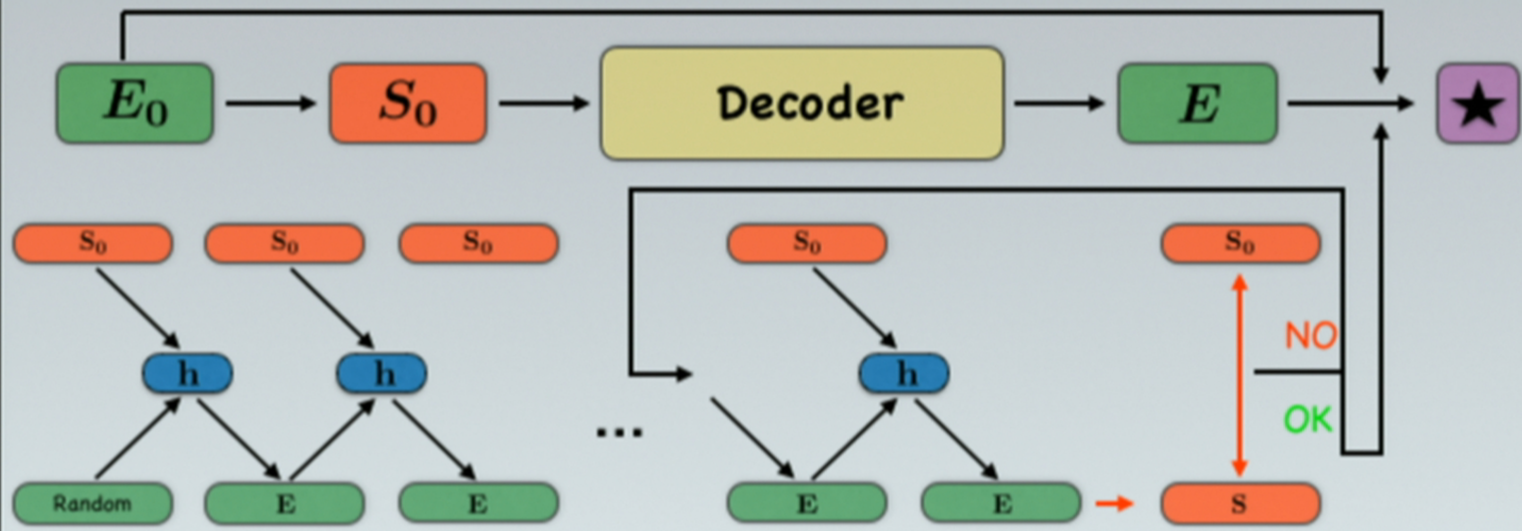
The Decoder



gtorlai@uwaterloo.ca

Machine Learning Decoder

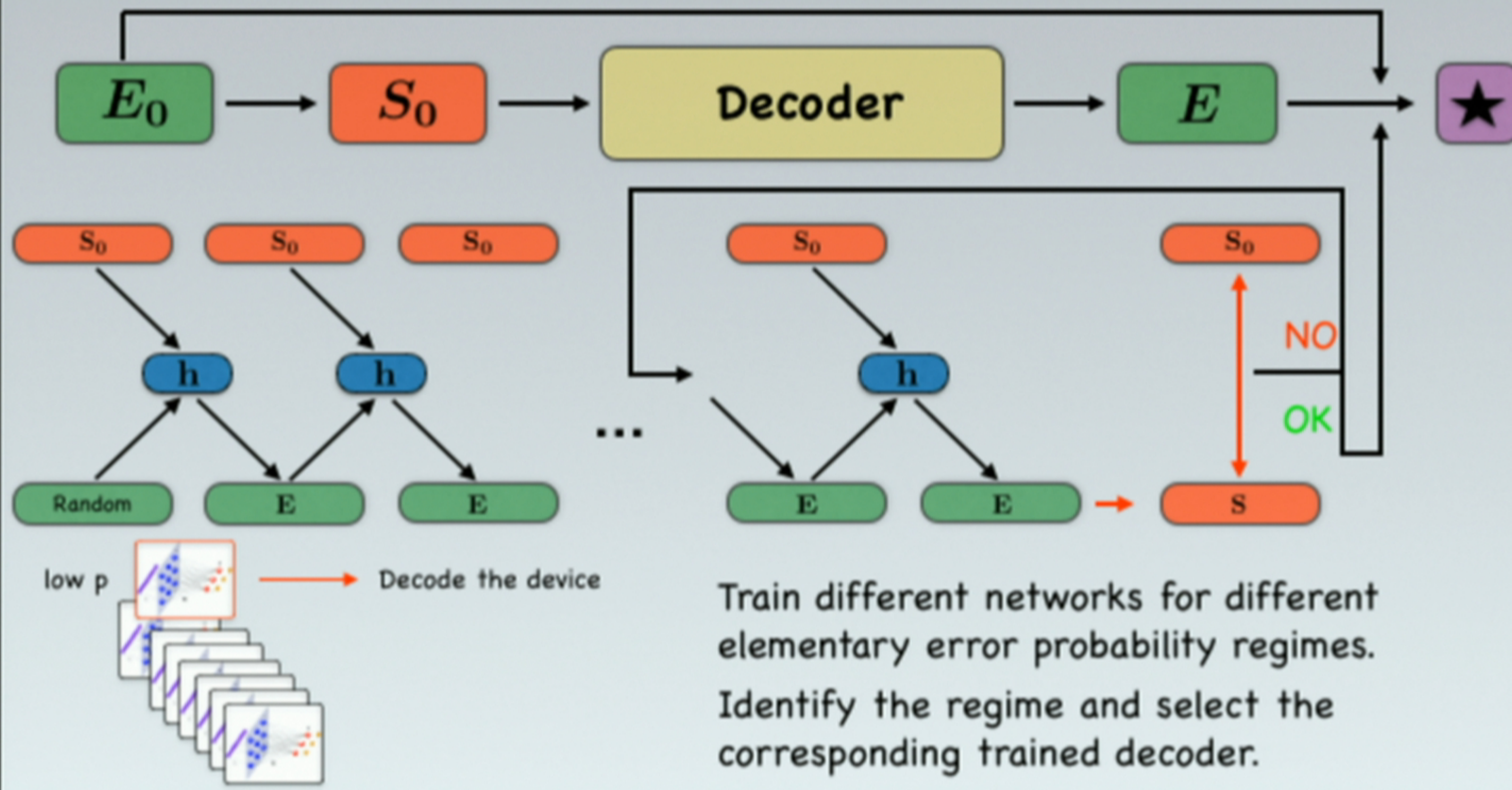
The Decoder



gtorlai@uwaterloo.ca

Machine Learning Decoder

The Decoder



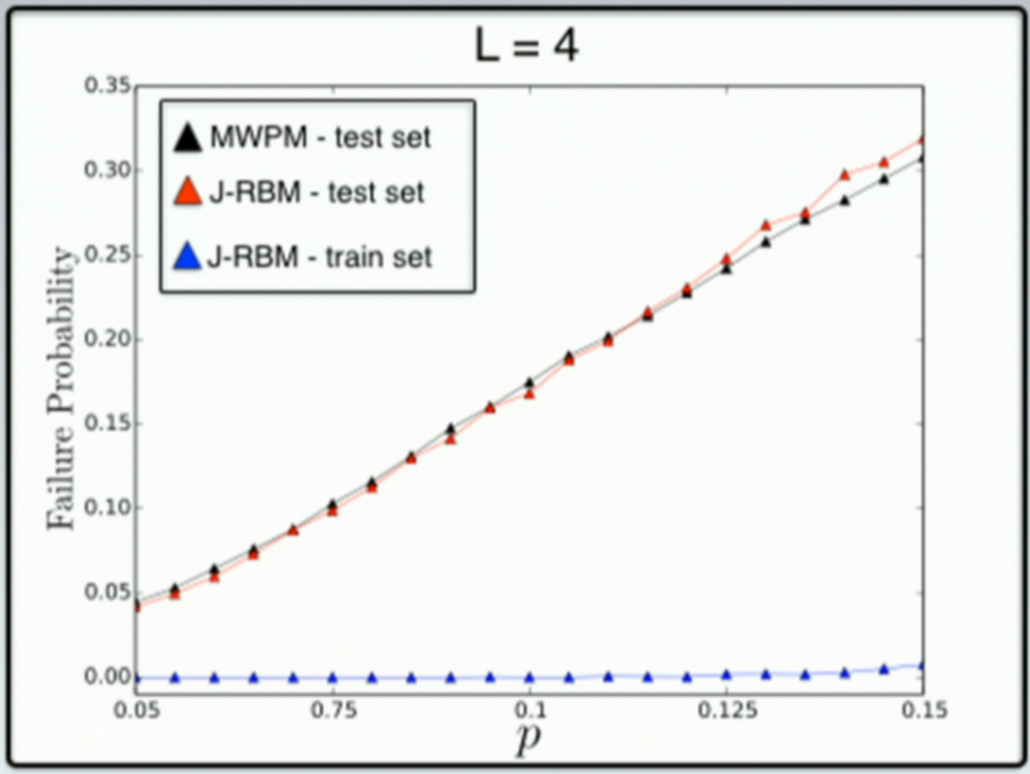
Train different networks for different elementary error probability regimes.

Identify the regime and select the corresponding trained decoder.

gtorlai@uwaterloo.ca

Machine Learning Decoder

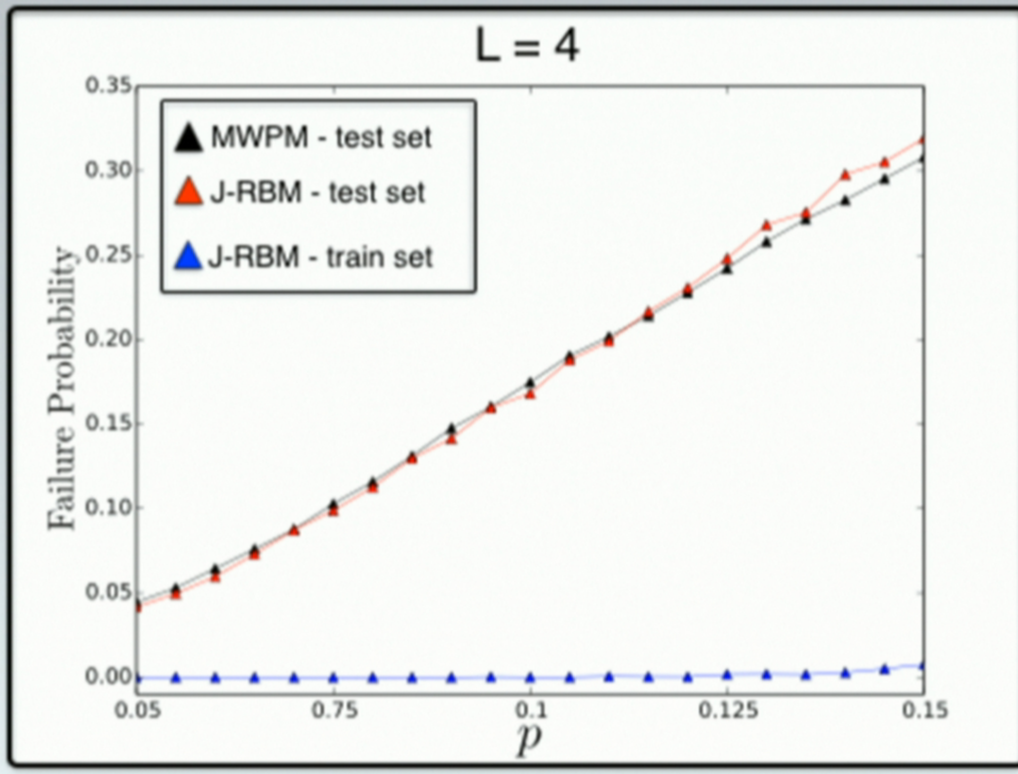
Decoding Accuracy: J-RBM vs MWPM



gtorlai@uwaterloo.ca

Machine Learning Decoder

Decoding Accuracy: J-RBM vs MWPM



gtorlai@uwaterloo.ca