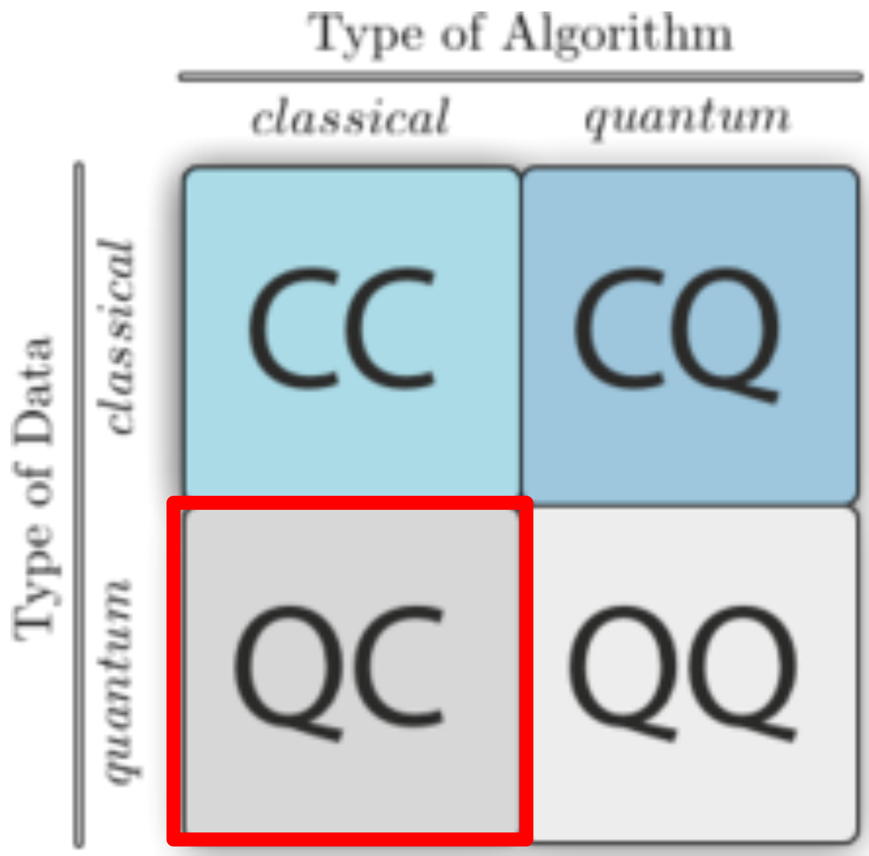
Solving the Quantum Many-Body Problem with Artificial Neural Networks

Carleo, Giuseppe, and Matthias Troyer. *Science* 355, no. 6325 (2017): 602-606.

Presented By: Remmy A. M. Zen
CQT Quantum Machine Learning Journal Club

NUS
National University of Singapore

Paper in a Nutshell



Paper in a Nutshell

Solving the Quantum Many-Body Problem with Artificial Neural Networks

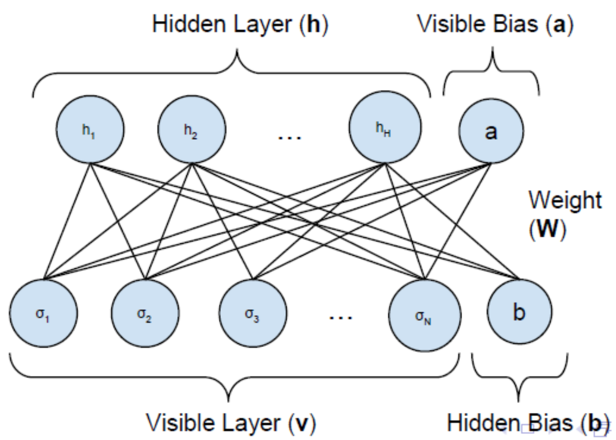
1. Ground State

2. Unitary Dynamics

$$|\Psi(s)|^2 = P(v)$$

Neural Network Quantum States (NQS)

Restricted Boltzmann Machine (RBM)



Roadmap

NUS National University of Singapore

Introduction → NQS → Experiments → Conclusion

- Quantum Physics
- Machine Learning

Roadmap

Introduction → NQS → Experiments → Conclusion

- **Quantum Physics**
- Machine Learning

Wave Function

$$\Psi$$

NUS National University of Singapore

- **Quantum many-body system** is a system made of more than two microscopic interacting particles.
- **Wave function** (Ψ) is used to describe the quantum state of a system.
- $|\Psi|^2$ gives you the probability density for each configuration S in the system.
- Impossible to save a complete wave function
  - 46 qubits = 2^{46} ~ 1 petabyte of memory!

Many-Body Problem

NUS National University of Singapore

Hamiltonian (*H*) is an operator corresponding to total **energy** of the system.

Central goal of Quantum mechanics to **solve the Schroedinger equation**

$$H|\Psi_i\rangle = E_i|\Psi_i\rangle$$

Find the smallest $\mathbf{E_0}$ **(ground-state energy)** corresponding to the $\mathbf{\Psi_0}$ **(ground-state wave function)**

How to represent Ψ?

Previous Work

NUS National University of Singapore

- **Quantum Monte Carlo (QMC)**
  - **Stochastic:** Sample a finite number of configurations.
  - (-) Break for frustrated models or sign problem.
- **Matrix Product States (MPS)**
  - **Compression:** Find an efficient representations.
  - (-) Mostly 1D geometries and lattice problems.

This Paper proposed **Neural Network Quantum States (NQS)!**

Variational Theorem

Variational Theorem:

$$E[\psi] = \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle} \geq E_0$$

$E[\psi]$ is the expectation of the energy function, ψ is some arbitrary state and E_0 is the exact ground-state energy of the Hamiltonian H.

Optimization Problem:

$$\psi_0 = argmin_\psi E[\psi]$$

$$E[\psi] = \frac{\sum_v |\psi(v)|^2 (\sum_{v'} H_{v,v'} \frac{\psi(v')}{\psi(v)})}{\sum_v |\psi(v)|^2}$$

Involves sums/integrals over high-dimensional space, hard to compute analytically

Variational Monte Carlo

W. L. McMillan, Phys. Rev. **138**, A442 (1965)

National University of Singapore

$$E[\psi] = \frac{\sum_v |\psi(v)|^2 (\sum_{v'} H_{v,v'} \frac{\psi(v')}{\psi(v)})}{\sum_v |\psi(v)|^2}$$

If we have samples distributed according to $|\ \psi(v)\ |^2$,

$E[\psi]$ can be written as a statistical expectation value of the local energy E_{loc}

$$E[\psi] = \langle\langle E_{loc}(v) \rangle\rangle$$

$$E_{loc}(v) = \sum_{v'} H_{v,v'} \frac{\psi(v')}{\psi(v)}$$

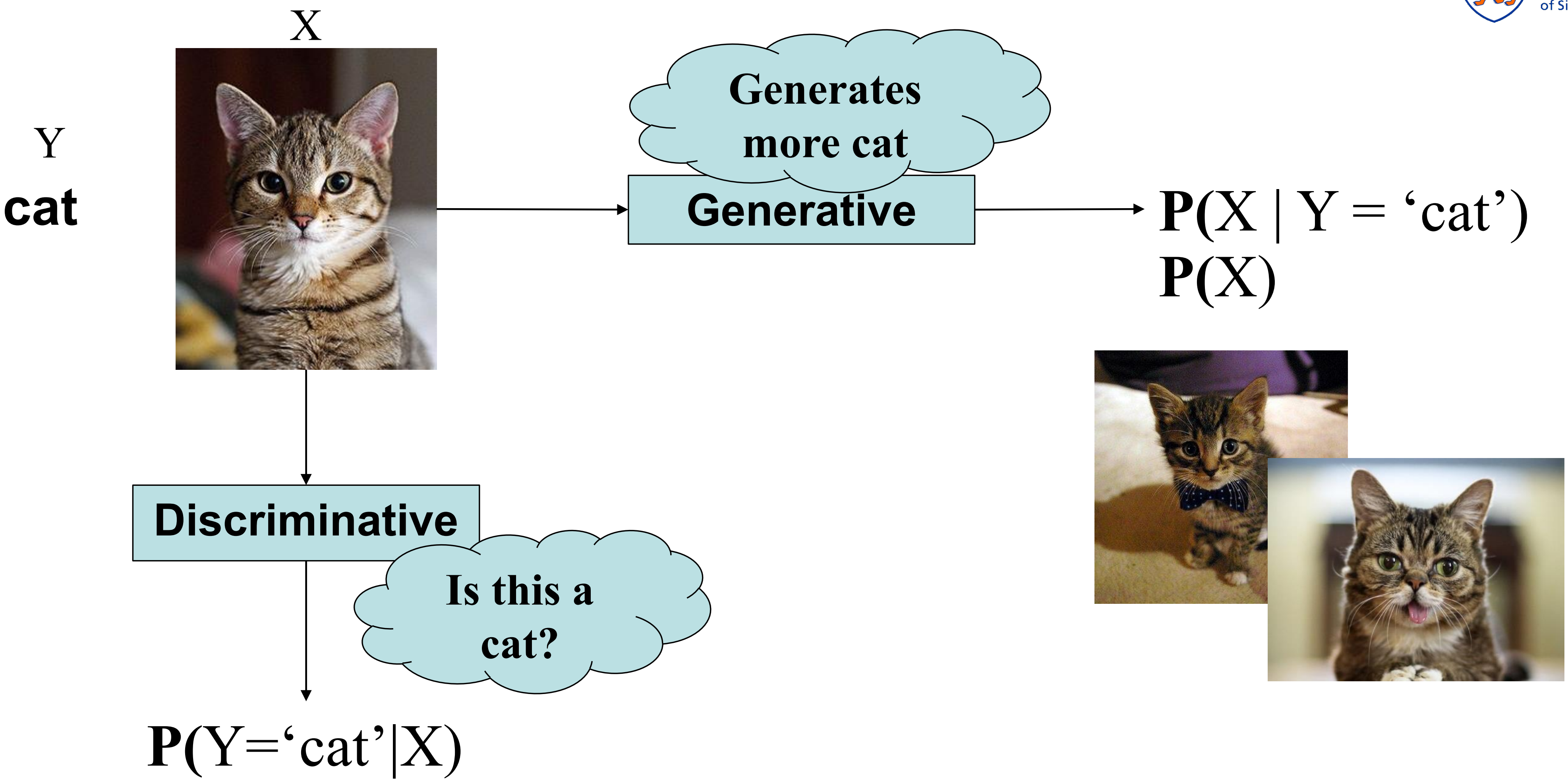
Roadmap

NUS National University of Singapore

Introduction → NQS → Experiments → Conclusion

- Quantum Physics
- **Machine Learning**

Generative vs Discriminative



Generative Model based on Neural Network

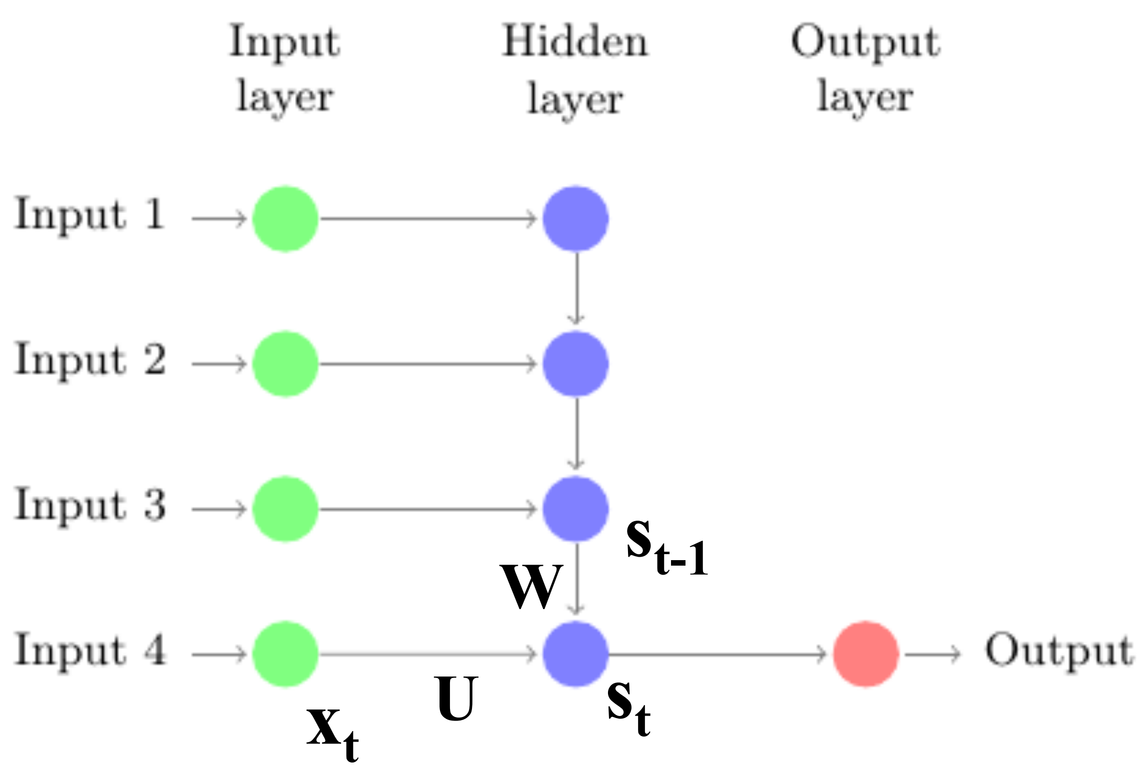
NUS National University of Singapore

- **Restricted Boltzmann Machine**
  - Initially 1986, Hinton at 2006 [1]
- **Recurrent Neural Network**
  - Initially 1980s, turned into generative at 2011 [2]
- **Variational Autoencoder (VAE)** [3]
- **Generative Adversarial Network (GAN)** [4]

[1] Hinton, G. E.; Salakhutdinov, R. R. (2006). "Reducing the Dimensionality of Data with Neural Networks“. Science. 313 (5786): 504–507.
[2] Sutskever, Ilya, James Martens, and Geoffrey E. Hinton. "Generating text with recurrent neural networks." Proceedings of the 28th International Conference on Machine Learning (ICML-11). 2011.
[3] Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013).
[4] Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems. 2014.

Recurrent Neural Network

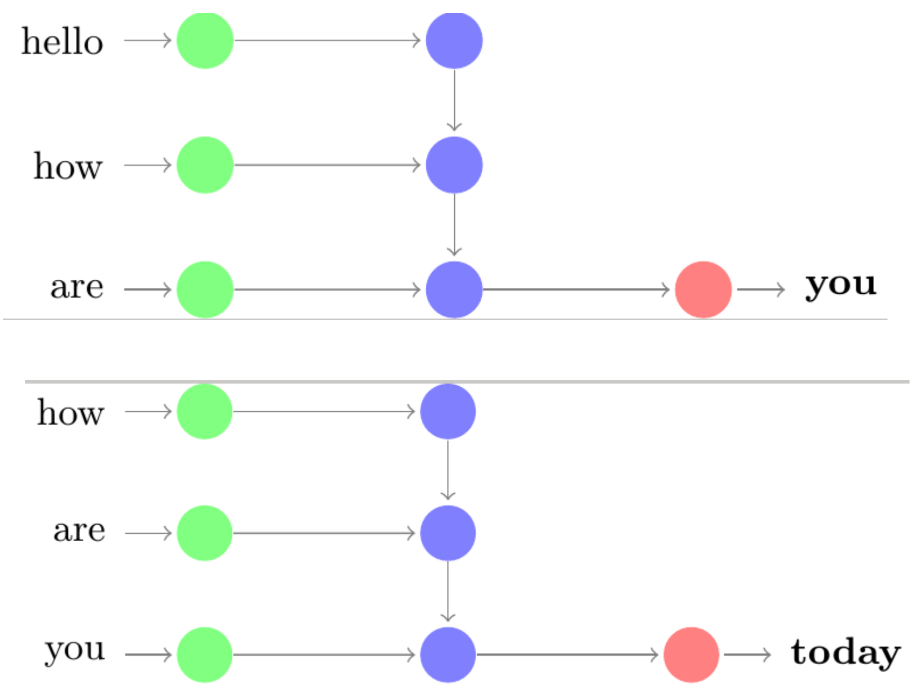
Neural network with recurrent unit. Ideal for Sequence data.

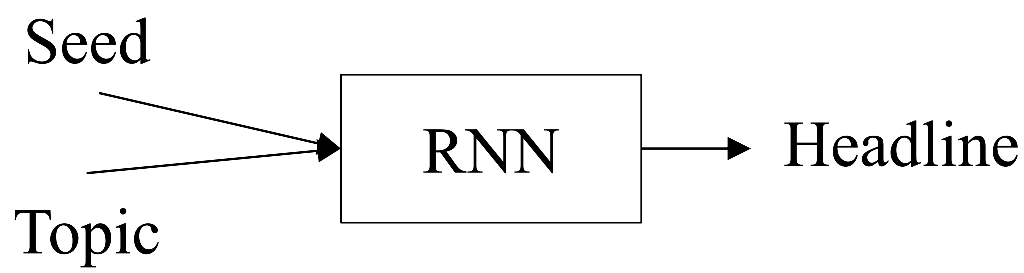


$$s_t = f(U x_t + W s_{t-1})$$

$$P(X_n \mid X_{n-1}, X_{n-2}, \ldots)$$

Recurrent Neural Network for Text Generation



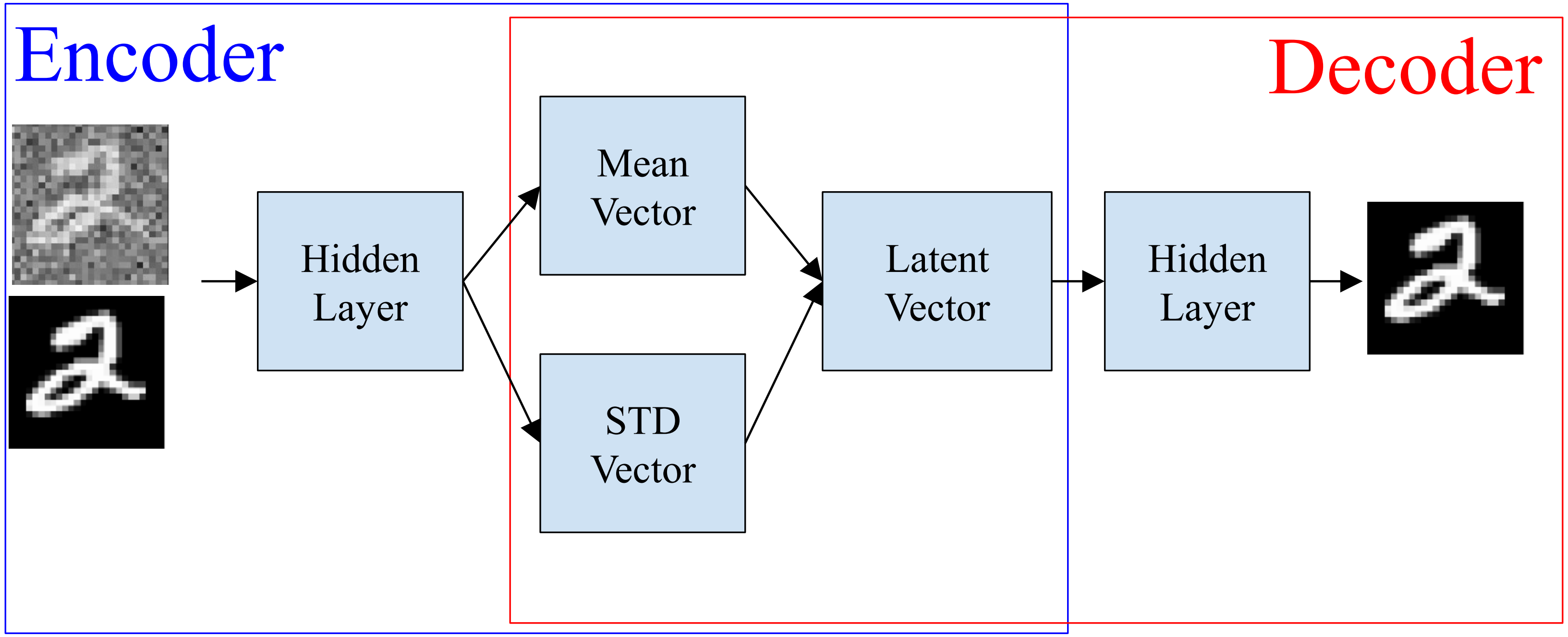


- ***Justin bieber apologizes*** **for racist joke on twitter for gay fans**
- ***Obama says that*** **he was kicked out of kim kardashian wedding**

Dandekar, Ashish, Remmy AM Zen, and Stéphane Bressan. "Generating Fake but Realistic Headlines Using Deep Neural Networks." *International Conference on Database and Expert Systems Applications*. Springer, Cham, 2017.

Variational Autoencoder (VAE)

NN where input is the same as output



Encoder: learn a latent vector
Decoder: reconstruct from latent vector

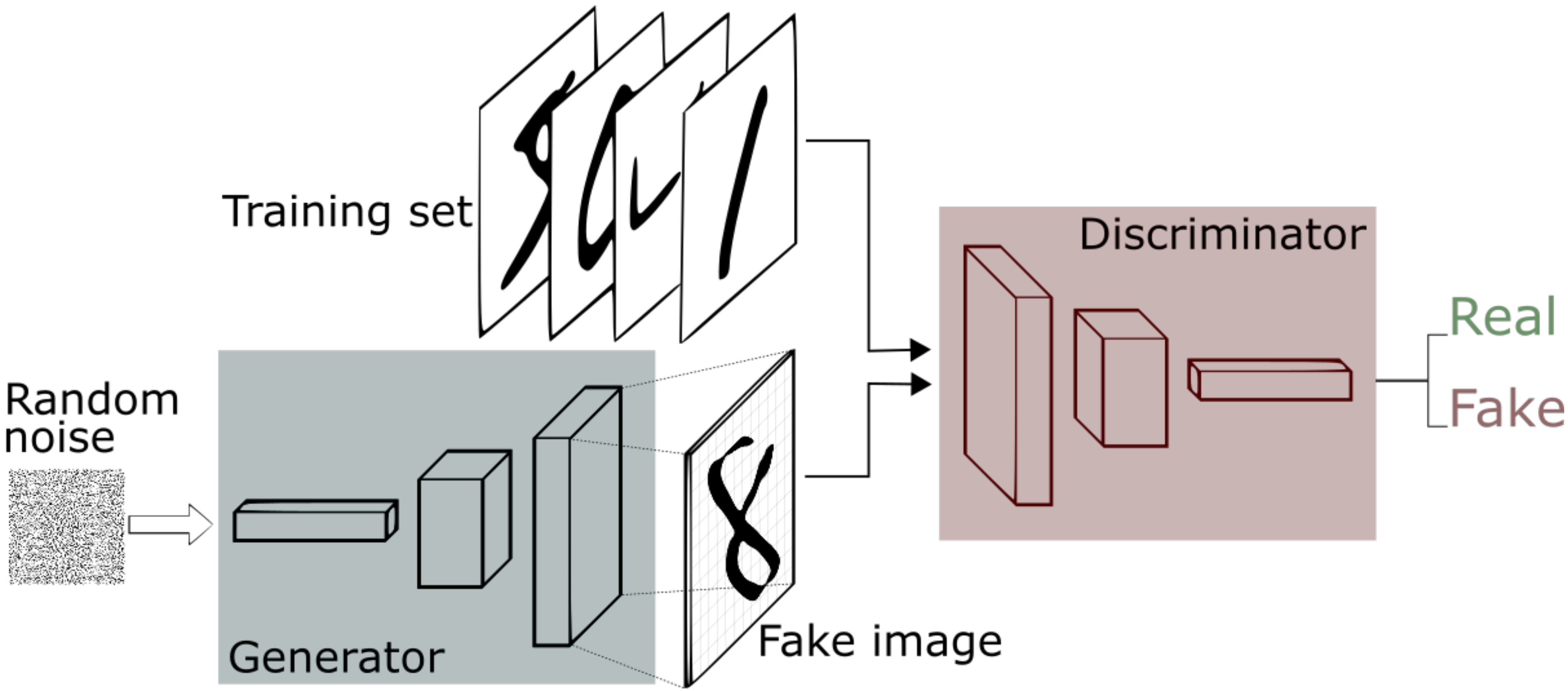
Generated Digits

Generated Faces

Exploring The Latent Vectors

Generative Adversarial Network (GAN)

Based on Game Theory, compete NN with another NN



Generative Adversarial Network (GAN)

State-of-the-art, able to produce high quality images.

"Most interesting idea in machine learning in last ten years"

> 400 models, check-out the GAN Zoo!

https://github.com/hindupuravinash/the-gan-zoo

GAN C-GAN DAGAN 3D-GAN FakeGAN

BiGAN VAE-GAN VariGAN

InfoGAN DRAGAN ABC-GAN

NUS National University of Singapore

“GANs are great, but we should go back to exploring more widely the landscape of generative models, such as ... **Boltzmann Machines.**”

— **Yoshua Bengio,** at ICML workshop on Theoretical Foundations and Applications of Deep Generative Models

Restricted Boltzmann Machine (RBM)

National University of Singapore

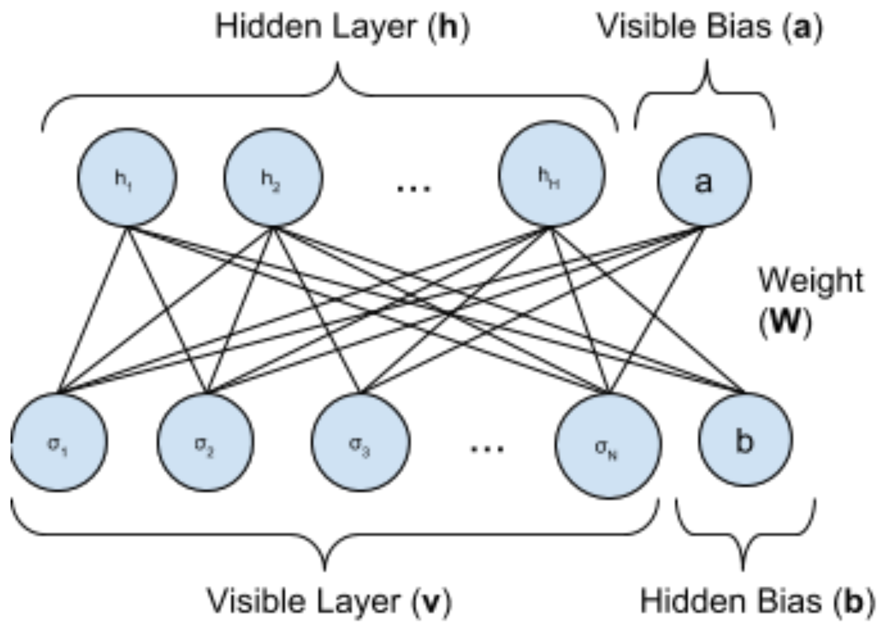
RBM is an undirected graphical model consists of:

- One visible layer (**v**) corresponding to physical spin variable

  $v = \sigma_1, \sigma_2, ..., \sigma_N, \sigma_i \in \{-1, +1\}$

- One hidden layer (**h**) corresponding to auxiliary spin variable

  $h = h_1, h_2, ..., h_{N_h}, h_i \in \{-1, +1\}$



Three parameters:

- weight connecting two layers (W)
- visible bias (a)
- hidden bias (b)

Restricted Boltzmann Machine (RBM)

NUS National University of Singapore

RBM is an energy based model, joint probability is given by the Boltzmann Distribution

$$p(\mathbf{v}, \mathbf{h}) = \frac{e^{-E(\mathbf{v},\mathbf{h})}}{Z}$$

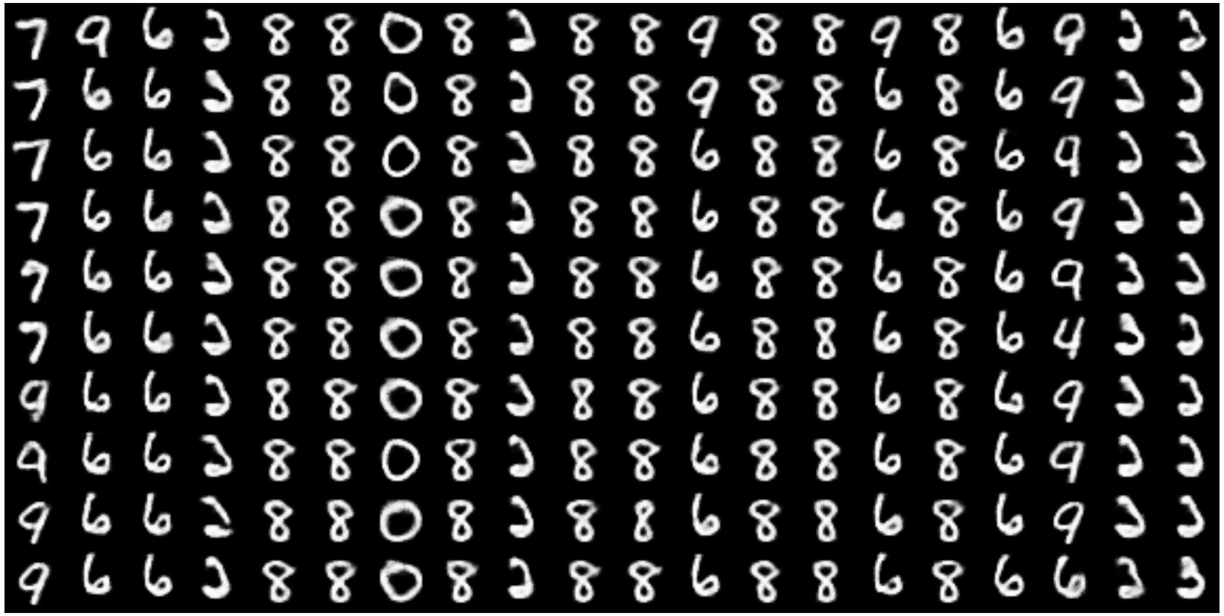
Where $E(\mathbf{v}, \mathbf{h}) = -\mathbf{a}^T\mathbf{v} - \mathbf{b}^T\mathbf{h} - \mathbf{v}^T\mathbf{W}\mathbf{h}$ and Z is the normalizing partition function.

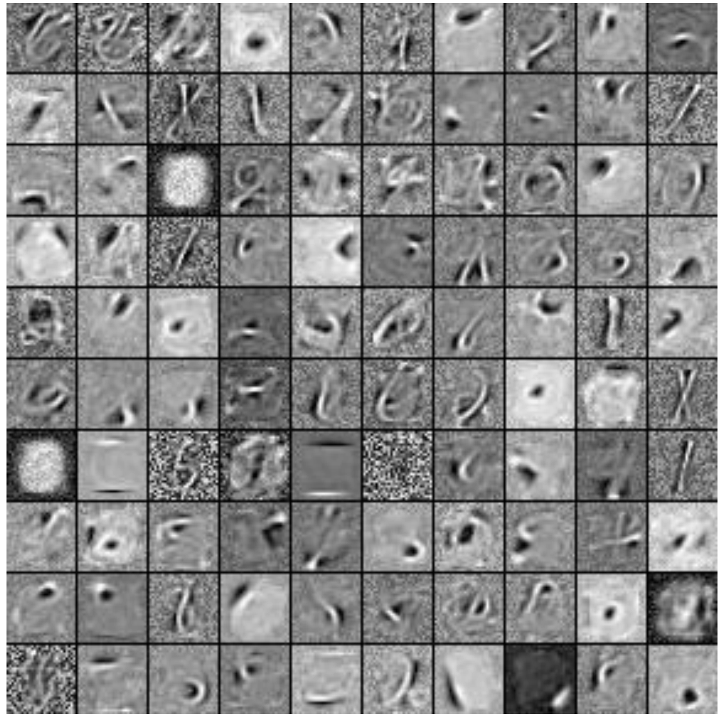
However, we are interested in p(**v**) so, we can marginalize h:

$$p(\mathbf{v}) = \frac{1}{Z} e^{\sum_{i=1}^{N} a_i \sigma_i} \prod_j 2cosh(b_j + \sum_{i=1}^{N} w_{ij} \sigma_i)$$

Restricted Boltzmann Machine (RBM)

Generated samples from Digit dataset





Roadmap

Introduction → **NQS** → Experiments → Conclusion

- Quantum Physics
- Machine Learning

W. L. McMillan, Phys. Rev. **138**, A442 (1965)

NUS National University of Singapore

Variational Monte Carlo

$$E[\psi] = \frac{\sum_v |\psi(v)|^2 (\sum_{v'} H_{v,v'} \frac{\psi(v')}{\psi(v)})}{\sum_v |\psi(v)|^2}$$

If we have samples distributed according to | ψ(v) |^2,

E[ψ] can be written as a statistical expectation value of the local energy E_{loc}

$$E[\psi] = \langle\langle E_{loc}(v) \rangle\rangle$$

$$E_{loc}(v) = \sum_{v'} H_{v,v'} \frac{\psi(v')}{\psi(v)}$$

Neural Network Quantum States (NQS)

NUS National University of Singapore

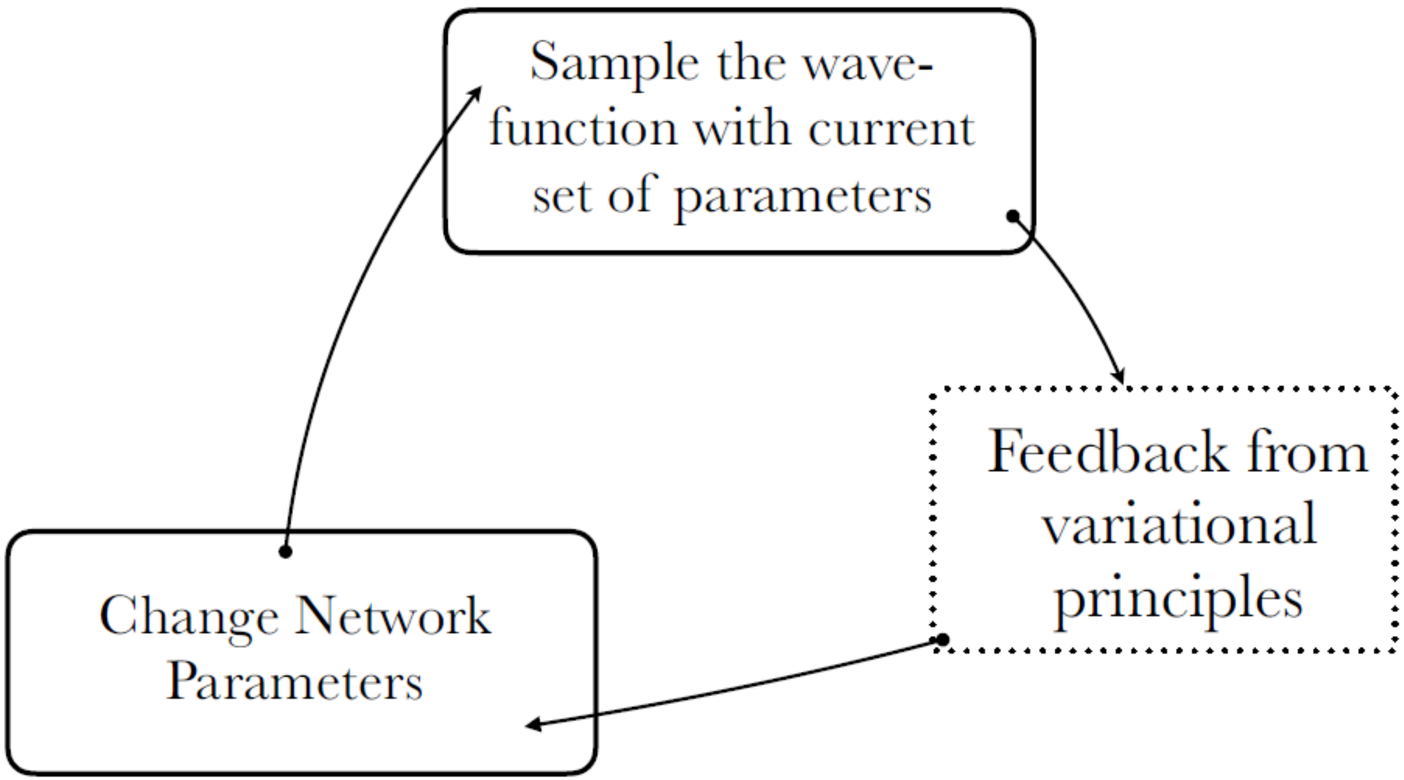
$$|\psi(v)|^2 = p(\mathbf{v})$$

$$\psi(v) = \sqrt{p(\mathbf{v})}$$

*Assume positive definite so modulus part can be ignored

It means that sampling p(v) using RBM is the same as sampling from | ψ(v) |^2!

Neural Network Quantum States (NQS)



Roadmap

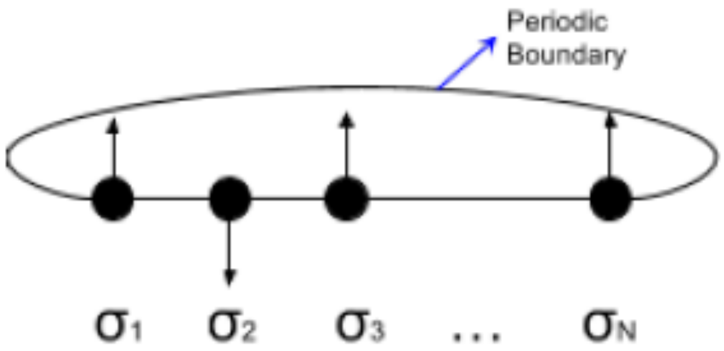
NUS National University of Singapore

Introduction → NQS → **Experiments** → Conclusion

- Quantum Physics
- Machine Learning

Experiments

Validate results with 1D Transverse Field Ising (TFI) model with Periodic Boundary Conditions



$$\mathcal{H}_{\mathrm{TFI}} = -h \sum_i \sigma_i^x - \sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z$$

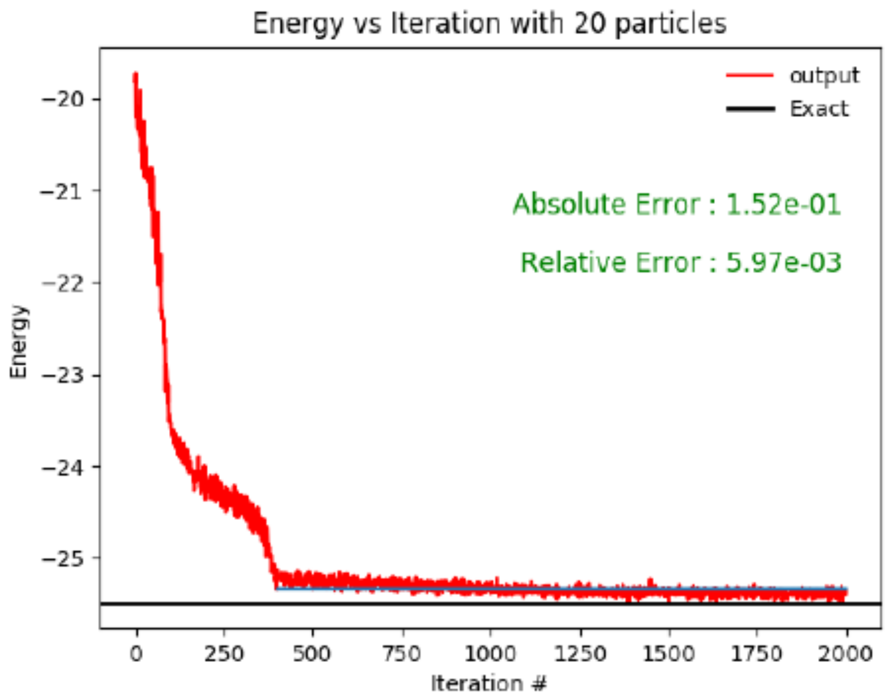
1D & 2D Antiferromagnetic Heisenberg (AFH) model.

$$\mathcal{H}_{\mathrm{AFH}} = \sum_{\langle i,j \rangle} \sigma_i^x \sigma_j^x + \sigma_i^y \sigma_j^y + \sigma_i^z \sigma_j^z,$$

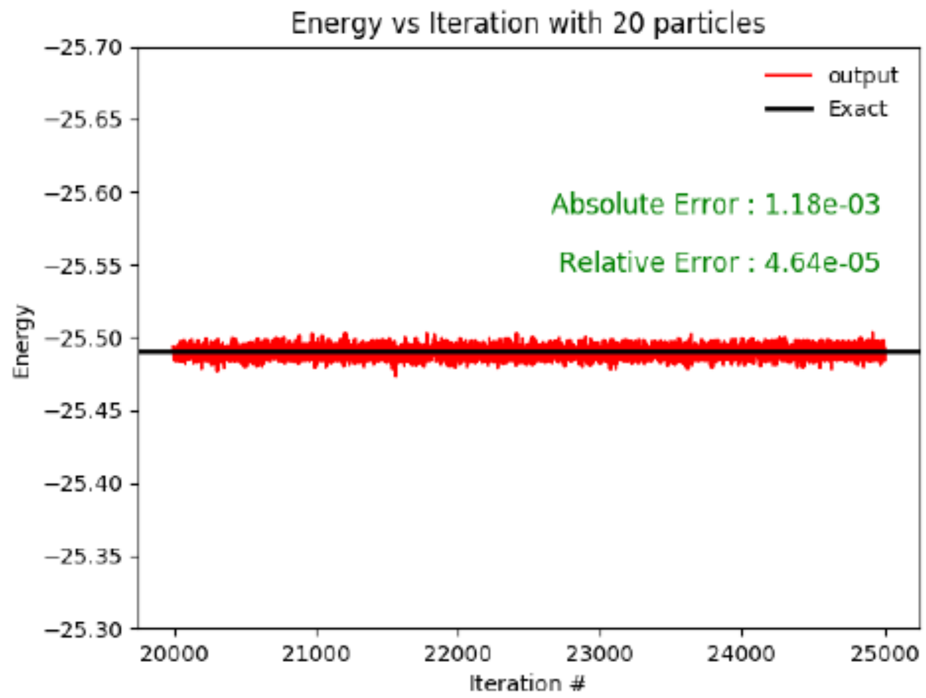
Results for TFI

NUS National University of Singapore

Compare with exact result from fermionization of TFI model.



(a) Energy and Error after 2000 iterations

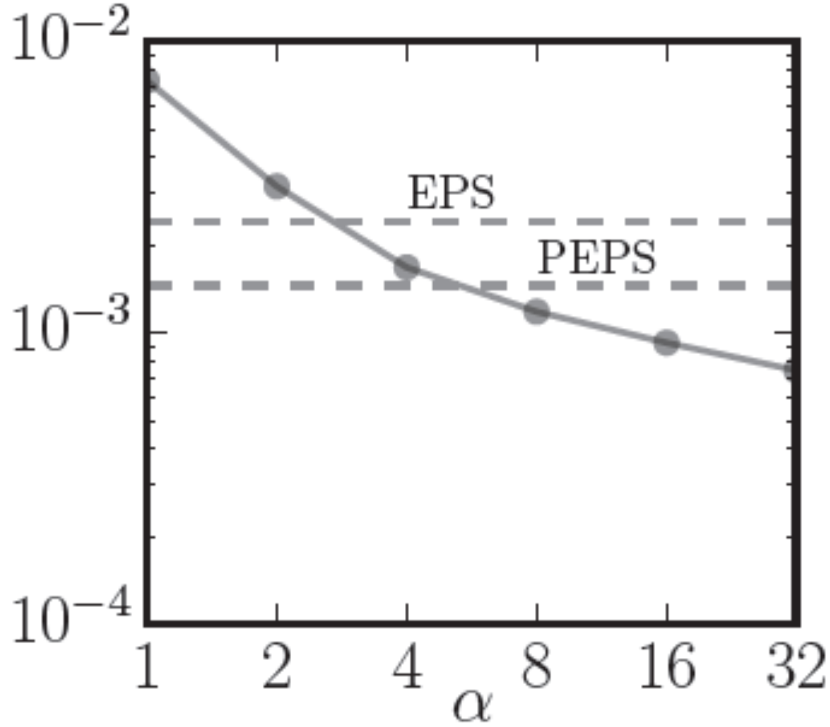


(b) Energy and Error after 25,000 iterations

Results for AFH

NUS National University of Singapore

Better than results from MPS when you increased the size of hidden nodes.

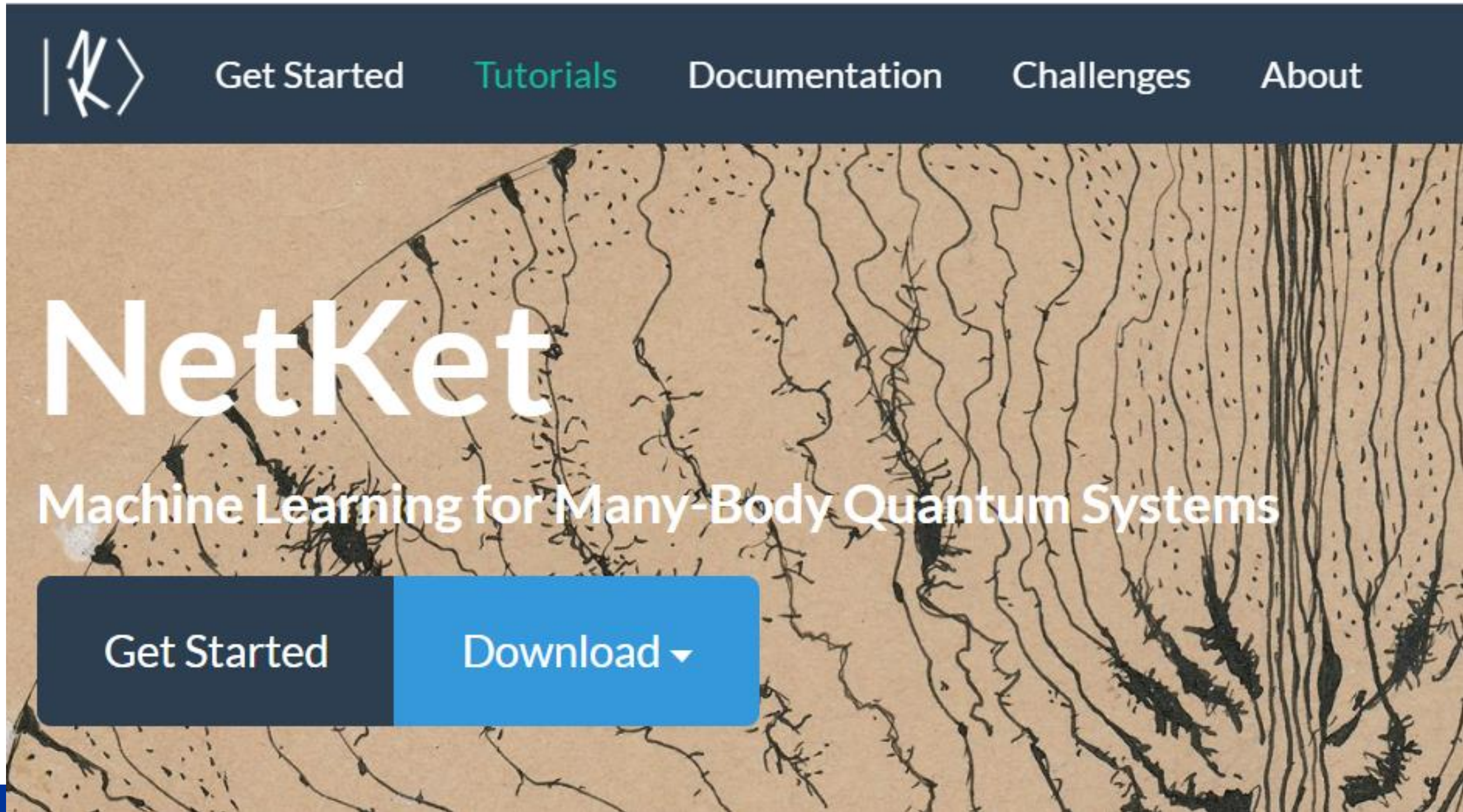


NetKet

National University of Singapore

https://www.netket.org/

A library by Giuseppe Carleo with C++ and MPI



NetKet

NUS National University of Singapore

https://www.netket.org/

A library by Giuseppe Carleo with C++ and MPI

Graphs
- Built-in Graphs
- Custom Graphs

Hamiltonians
- Built-in Hamiltonians
- Graph Hamiltonians
- Custom Hamiltonians

Learning
- Introduction
- Optimizers
- Learning the Ground State

Machines
- Introduction
- Restricted Boltzmann Machine
- Feedforward Neural Networks

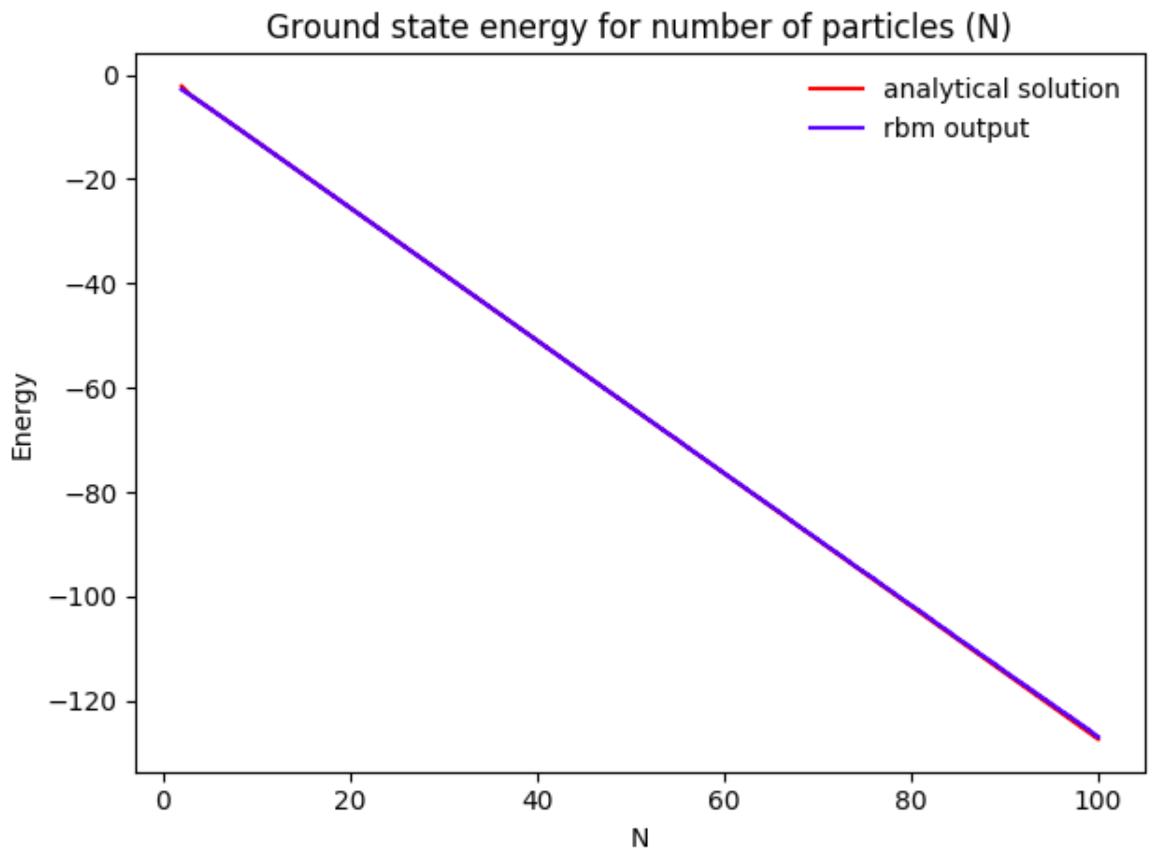
Observables
- Custom Observables

Sampling
- Introduction
- Local Moves
- Hamiltonian Moves

Experiment on Number of Particles

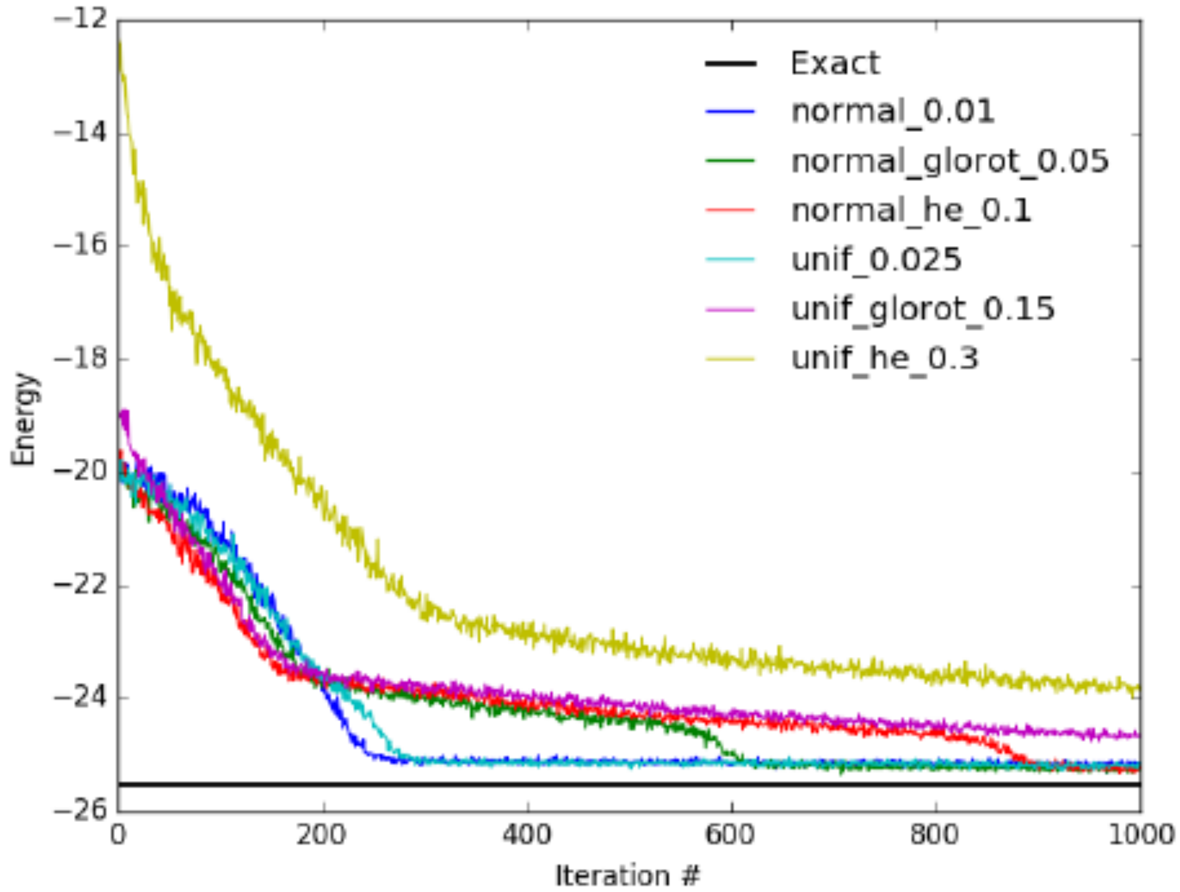
NUS National University of Singapore

We are currently implementing it with Tensorflow and we can find energy for 100 particles.

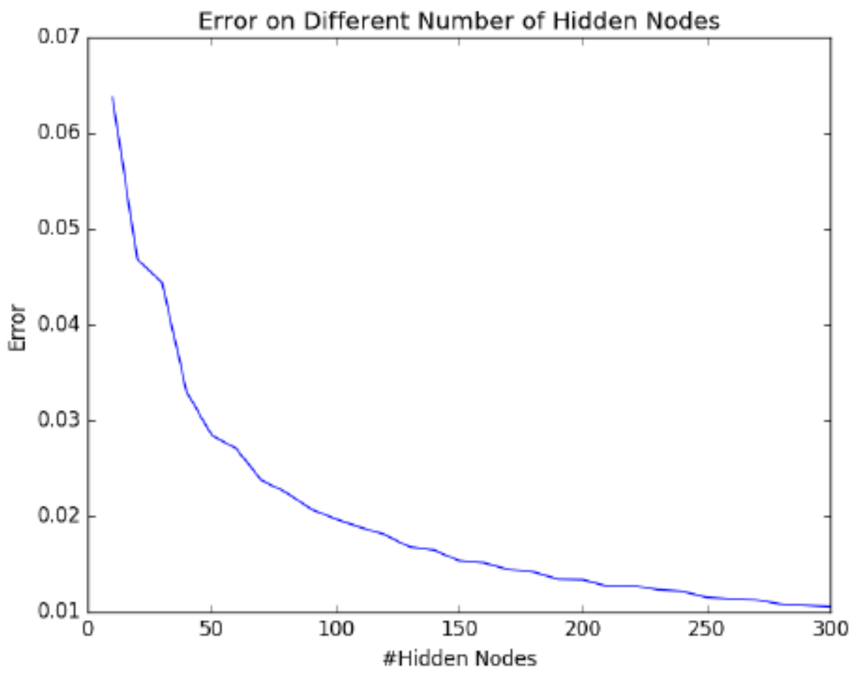


Experiments on Initializer for 1D TFI

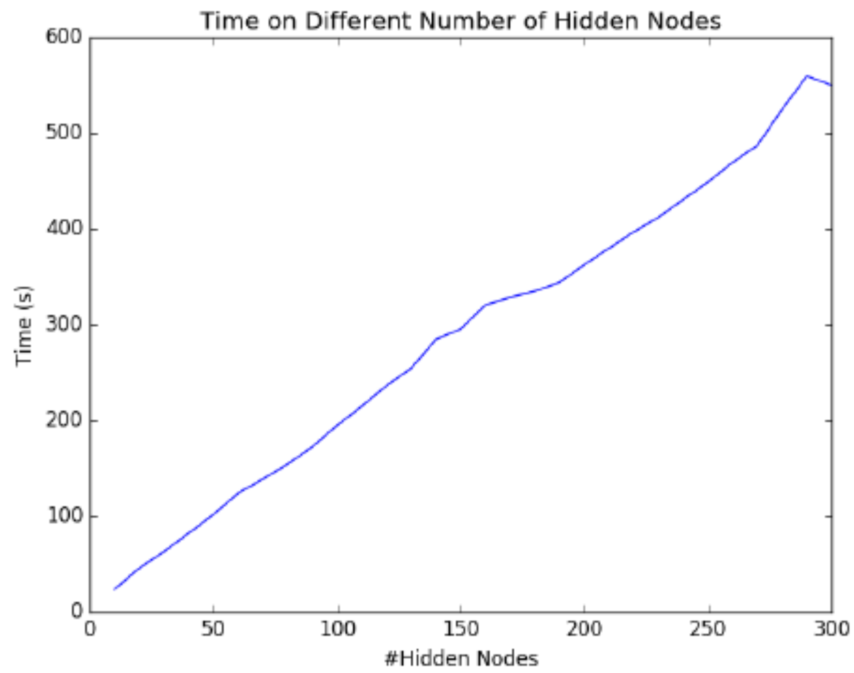
Method	Sampling
Hinton [4]	$\mathcal{N}(0, 0.01)$
Glorot Normal [2]	$\mathcal{N}(0, \frac{2}{v+h})$
He Normal [3]	$\mathcal{N}(0, \frac{2}{v})$
Glorot Uniform [2]	$\mathcal{U}(-\frac{6}{v+h}, \frac{6}{v+h})$
He Uniform [3]	$\mathcal{U}(-\frac{6}{v}, \frac{6}{v})$



Experiments on Number of Hidden Nodes for 1D TFI

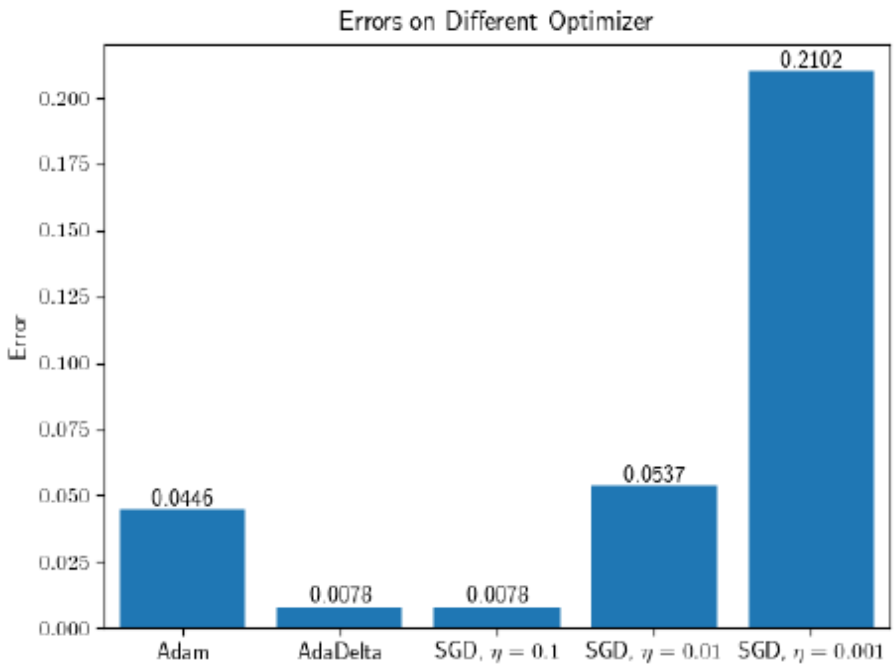


(a) Errors on Different Number of Nodes in Hidden Layer after 1000 epochs

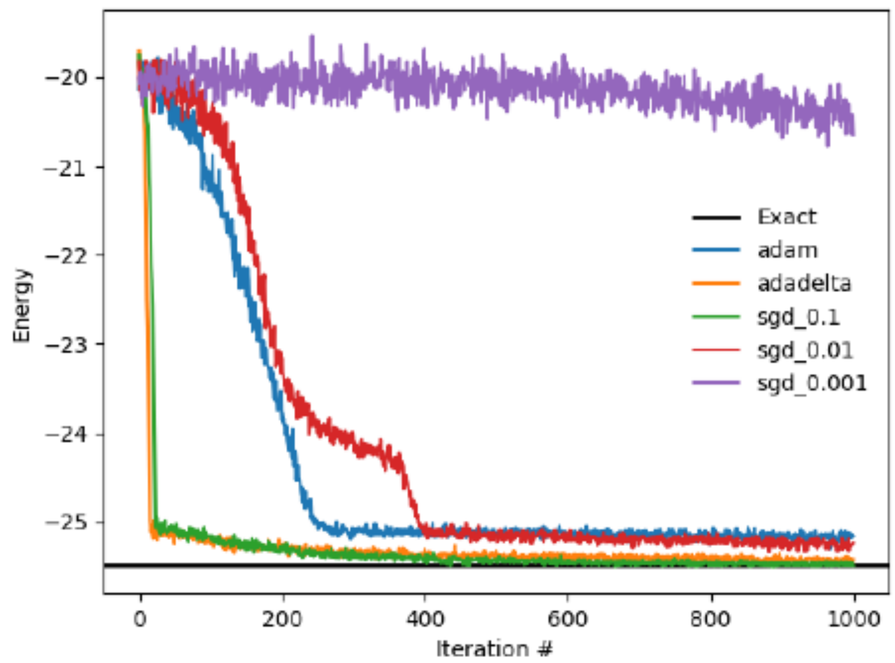


(b) Times on Different Number of Nodes in Hidden Layer after 1000 epochs

Experiments on Different Optimizer for 1D TFI



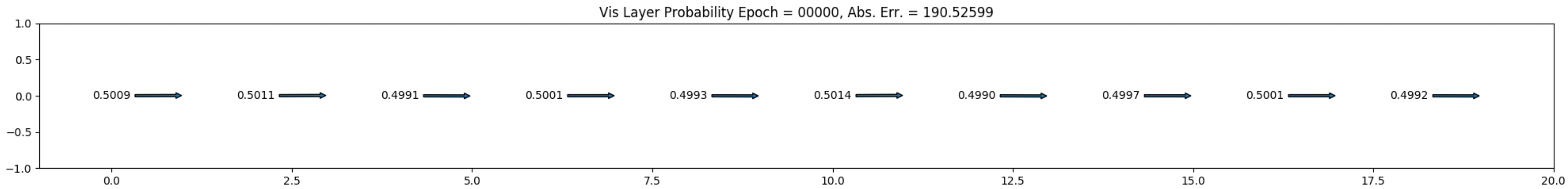
(a) Errors on Different Optimizer after 1000 epochs



(b) Lowest Energy w.r.t the Number of Iterations

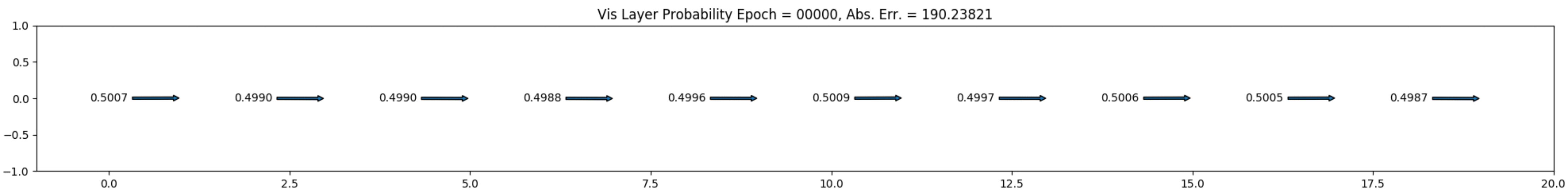
Visible Layer Visualization
1D TFI with h = 1, J = 1

$$H(x) = -h\sum_i \sigma_i - J\sum_i \sigma_i\sigma_{i+1}$$



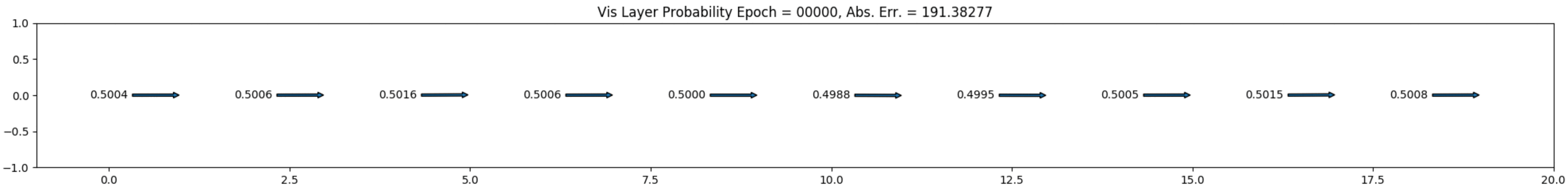
Visible Layer Visualization
1D TFI with h = 1, J = 10

$$H(x) = -h\sum_i \sigma_i - J\sum_i \sigma_i\sigma_{i+1}$$



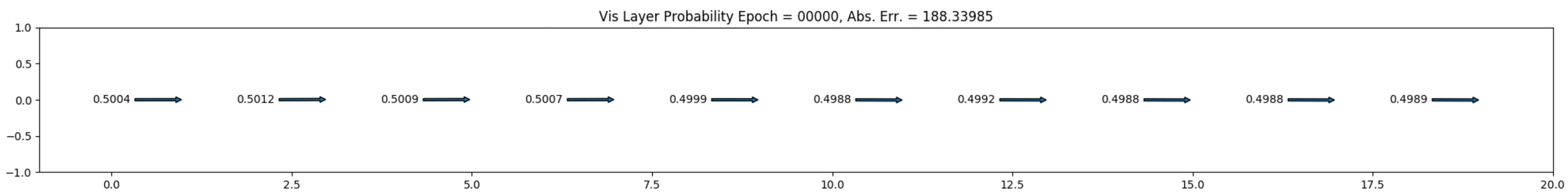
Visible Layer Visualization
1D TFI with h = 1, J = 10

$$H(x) = -h\sum_i \sigma_i - J\sum_i \sigma_i\sigma_{i+1}$$



Visible Layer Visualization
1D TFI with h = 1, J = 10

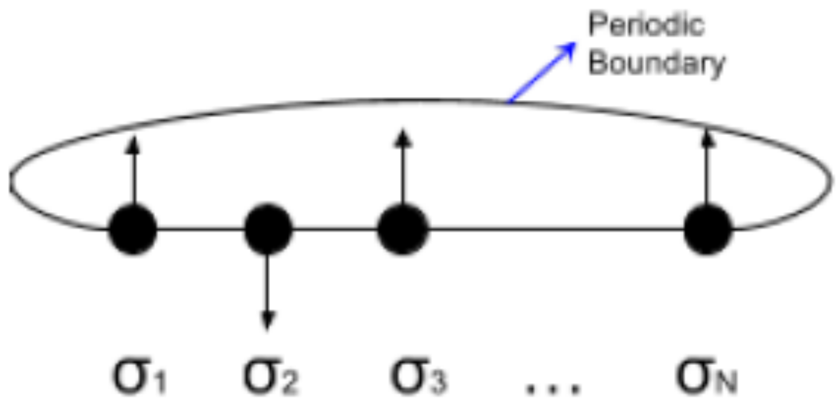
$$H(x) = -h\sum_i \sigma_i - J\sum_i \sigma_i\sigma_{i+1}$$

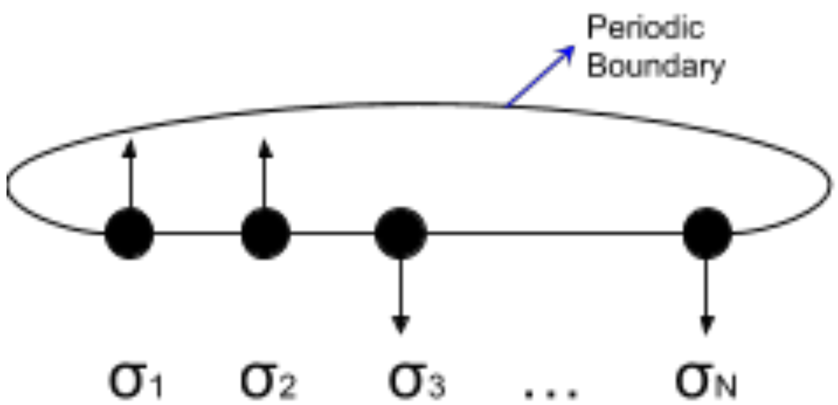


Symmetry Constraint

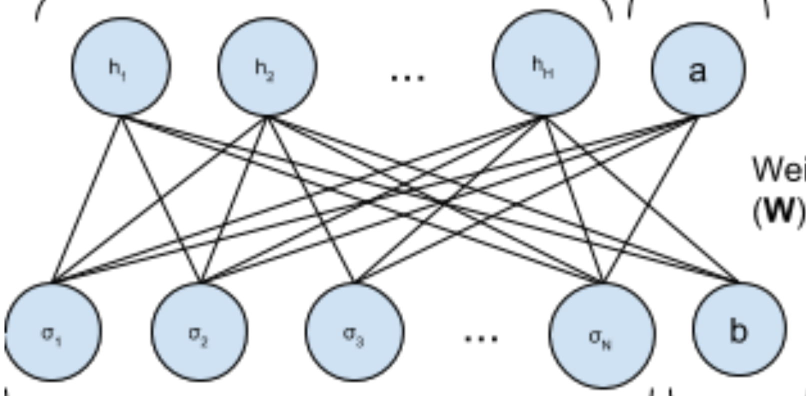
National University of Singapore

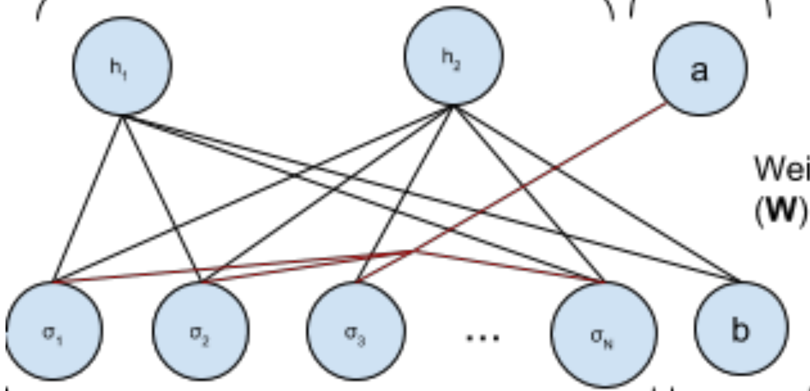
There is a lattice translation symmetry in the system. Could we impose that?





Symmetry Constraint





$W = v * h$
$a = v$
$b = h$

$W = \alpha * v$ $\alpha = h / v$
$a = 1$
$b = \alpha$

“Harder” to Optimize

NUS National University of Singapore

Epoch to reach 0.001 error

Asymmetric: 1380.2 epochs

Symmetric: 1977.6 epochs

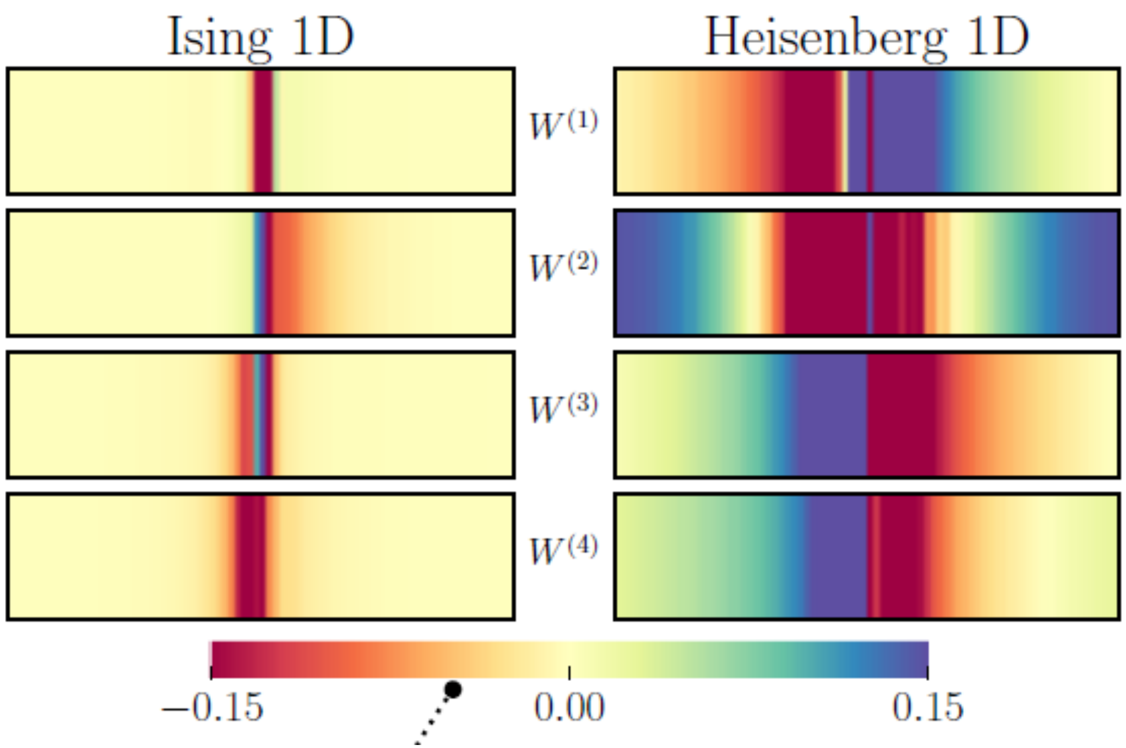
Energy after 1000 epochs

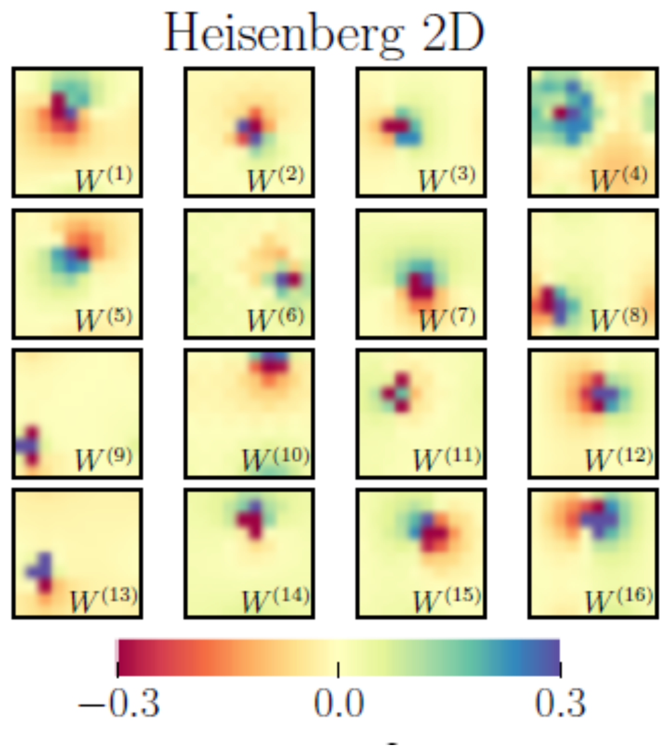
Asymmetric: -25.4817

Symmetric: -25.4057

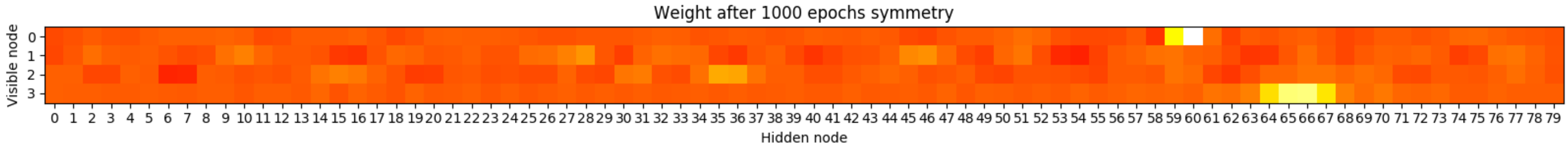
Exact: -25.4910

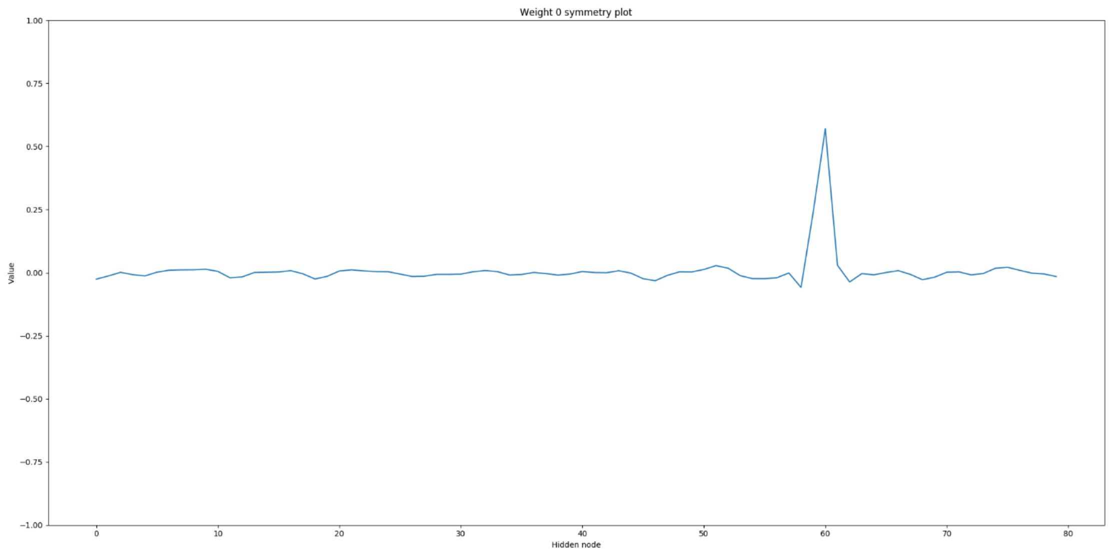
Weight Visualization

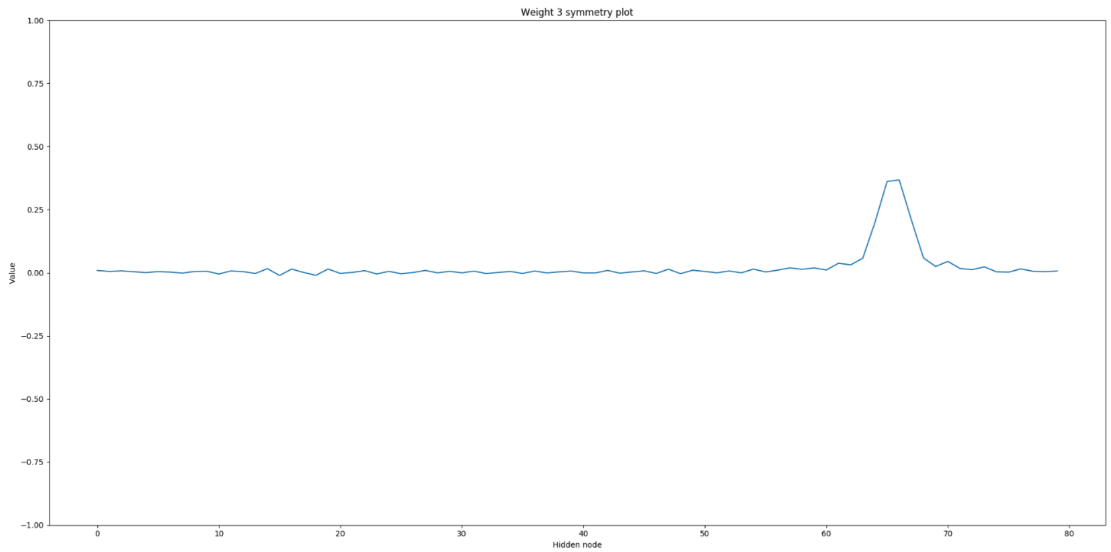




With the Tensorflow Code







Roadmap

NUS National University of Singapore

Introduction → Proposed Method → Experiments → **Conclusion**

Conclusion and Future Work

National University of Singapore

- **Proposed Neural Network Quantum States that can be used to solve Quantum Many Body Problem by using RBM to represent wave function**
- **Future Works:**
  - Deep Learning?
  - Other quantum systems other than spins.