Title: Learning quantum annealing

Date: Aug 09, 2016 02:30 PM

URL: http://pirsa.org/16080010

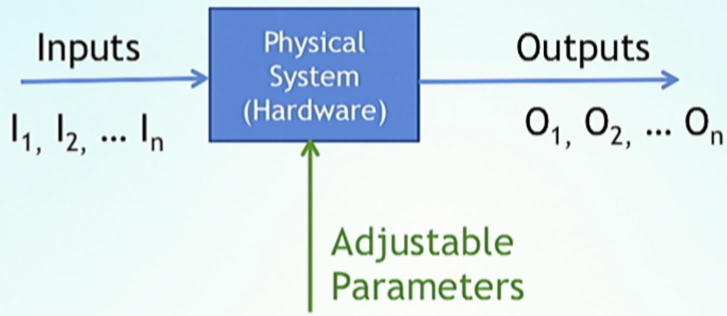
Abstract:

* Learning Entanglement: SQUID qubits and quantum annealing

James E Steck, Elizabeth C Behrman and Nam Nguyen
Wichita State University, Wichita, KS 67260
behrman@math.wichita.edu; james.steck@wichita.edu

With thanks to: A. Keister, M. Thom, T. Lanting (DWave), NASA USRA

Computing Hardware



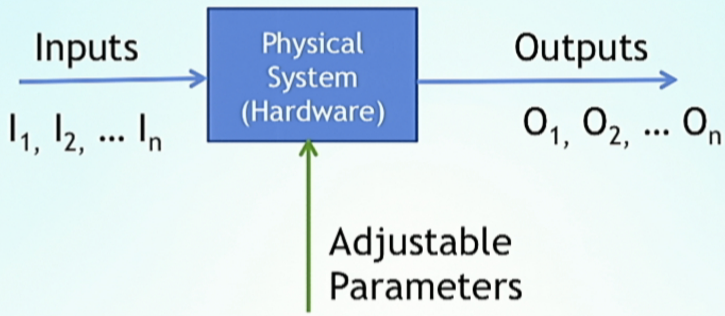
Any hardware with adjustable parameters and sufficient computing complexity can be made into a computation device

- Fluid flow in pipes+valves+flow constriction
- Ants
- Mechanical devices
- Biological Nervous System
- Electron Flow in Resistors+Capacitors+Inductors: Analog Computer; Hopfield Associative Network
- Transistors+Capacitors+etc on Silicon: Microprocessors
- Quantum Mechanics

*Quantum Computing

* Large scale implementation: always "ten years" away – because:
  - Algorithm construction
  - Scaleup
  - Noise
  - Decoherence
* Approaches
  - Decomposition into gates "circuit model"
  - Quantum Bits
  - Quantum annealing
  - **Machine Learning (This Workshop)**

Our Contributions to Quantum Machine Learning



Learning: Choosing Adjustable Parameters from examples to get correct I->O map

- Quantum Dots – learned 2 input classical and quantum logic gates (1996-2002)
- SQUIDS – learned 2, 3 qubit gates (2002-2007)
- SQUIDS – Learned Entanglement Witness for 2,3, 4, 5, 6 qubits (2002-2008)
- SQUIDS – Learned and corrected phase shifts in quantum states(2013)
- SQUIDS – Learned Entanglement Indicator robust to Noise, Decoherence (2015)
- Dwave – Learned to anneal to entangled states 2, 3, 4, 5, 6 qubits (2016)
- Quantum Lattice Gas – spatial continuum with soliton pathway behavior (2014)

Starting with the Schrödinger equation

$$\frac{\partial \rho}{\partial t} = \frac{1}{i\hbar}[H, \rho]$$

Where ρ is the density matrix and H is the Hamiltonian. For an N-qubit system, the Hamiltonian H is:

$$H = \sum_{i=1}^{N} K_\alpha \sigma_{x\alpha} + \epsilon_\alpha \sigma_{z\alpha} + \sum_{\alpha \neq \beta = 1}^{N} \zeta_{\alpha\beta} \sigma_{z\alpha} \sigma_{z\beta}$$

Where $\{\sigma\}$ are the Pauli operators corresponding to each qubits, $\{K\}$ are the tunneling amplitudes, $\{\epsilon\}$ are the biases, and $\{\zeta\}$ are qubit-qubit couplings.

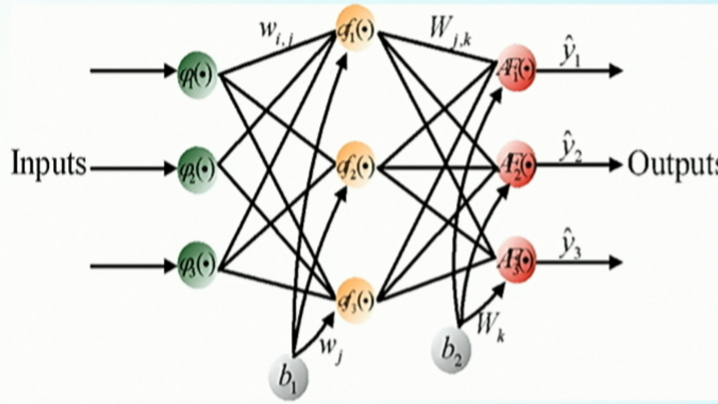
*Learn Entaglement Indicator

The general solution to the Schrodinger equation has a mathematical isomorphism to the equation for information propagation in a neural network

$$\rho(t) = e^{iHt/\hbar}\rho(0)e^{-iHt/\hbar}$$

$$\phi_i = \sum_j w_{ij} f_j(\phi_i)$$

$$\phi_{output} = F_W \phi_{input}$$



Where $\phi\downarrow output$ and $\phi\downarrow input$ are the output, and input vector of the networks.

$F\downarrow W$ is the network operator, which depends on the neuron connectivity weight matrix W.

QMLM Aug. 2016

For our system,

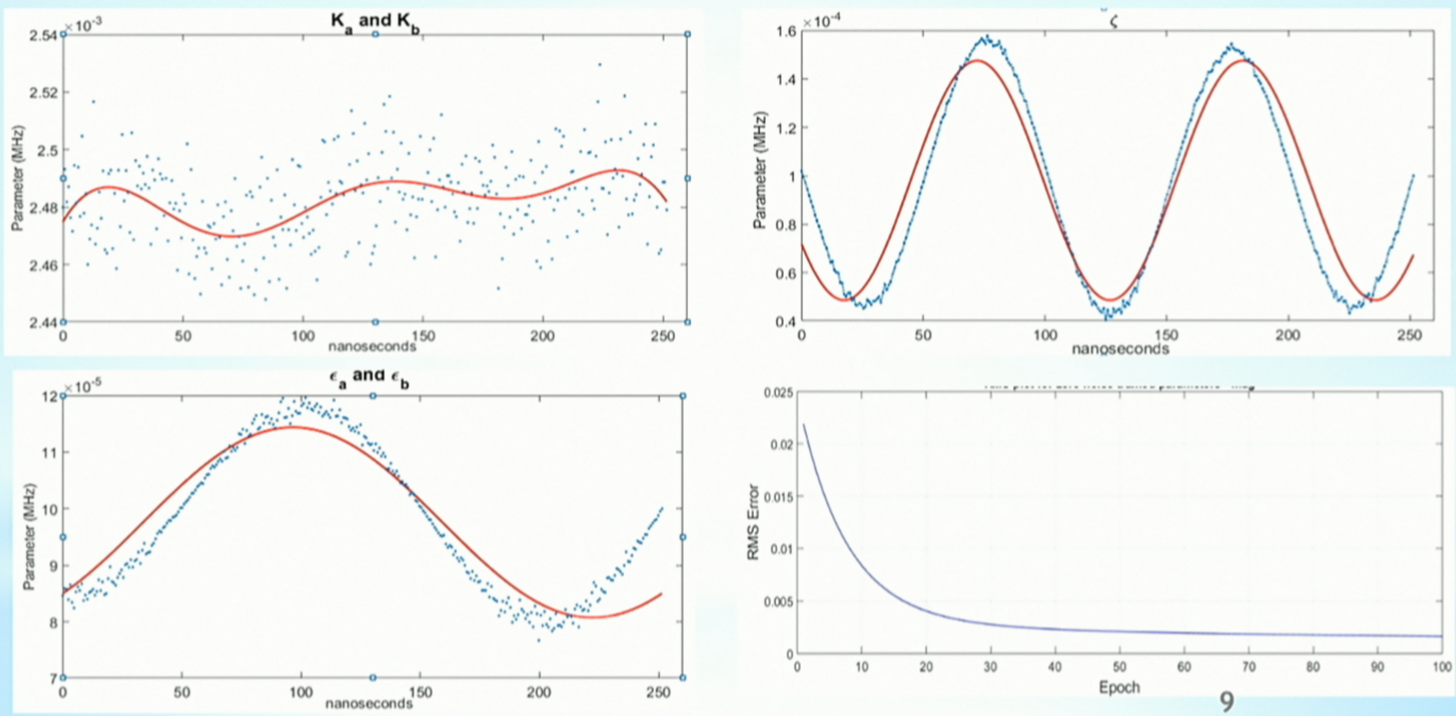
- * $\rho(0)$ = Input state vector
- * $\rho(t{\downarrow}f)$ = Output state vector
- * $\{K, \epsilon, \zeta\}$= Weights of the network, which can be adjusted experimentally as functions of time for the SQUID system under consideration.

Our goal is to train the external parameters $\{K, \epsilon, \zeta\}$ (weights) via **supervised learning** using target outputs paired with specified inputs

Once trained, the parameters can be tested on additional inputs.

QMLM Aug. 2016

Training with (magnitude) noise- parameters function



9

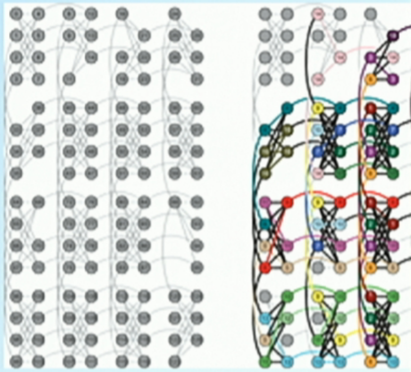
3,4, 5 and 6 qubit systems

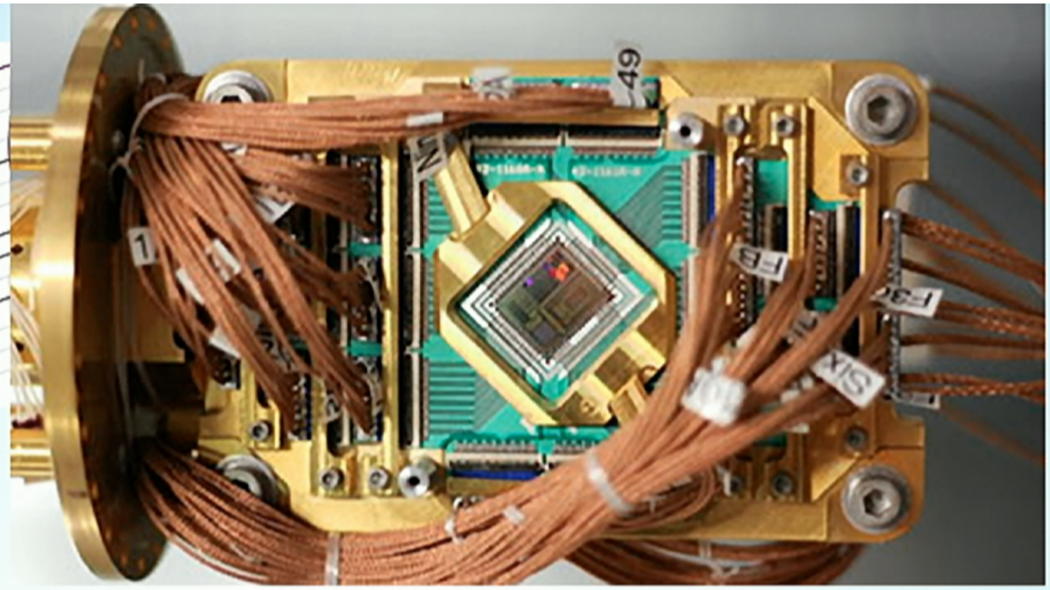
We observed similar results for the 3, 4, 5 and 6 qubit system. And, the system trains much better and the # of epochs decreases exponentially as we increased the number of qubits.

For a 3 qubit system, the Hamiltonian H is defined as

$$H = K_A\sigma_{xA} + K_B\sigma_{xB} + K_C\sigma_{xC} + \epsilon_A\sigma_{zA} + \epsilon_B\sigma_{zB} + \epsilon_C\sigma_{zC} + \zeta_{AB}\sigma_{zA}\sigma_{zB} + \zeta_{AC}\sigma z A\sigma_{zC} + \zeta_{BC}\sigma_{zB}\sigma_{zC}$$

Observe that the connectivity in the system increases by the order of





*D-Wave quantum annealing chip 1097 SQuID array

QMLM Aug. 2016

Given an initial state, $\rho(0)$, the system evolves in time according to the Schrödinger equation as:

$$\rho(t) = e^{iHt/\hbar}\rho(0)e^{-iHt/\hbar}$$

By analytically continue to imaginary time $t \rightarrow i\beta\hbar$:

$$\rho(\beta) = e^{-\beta H}\rho(0)e^{\beta H}$$

Now we use the interaction representation for the two “parts” of the Hamiltonian:

$$\rho_I = e^{-\beta H}\rho_S(t)\ e^{\beta H}$$

*Learning quantum annealing

QMLM Aug. 2016

* We define a Lagrangian, L, to be minimized:

$$L = \frac{1}{2}\left|\rho_{I,des} - \rho_I(t_f)\right|^2 + \int_0^{t_f} \lambda^+(t)\exp(-\beta H)\left(\frac{\partial \rho_S}{\partial t} + \frac{i}{\hbar}[H, \rho_S]\right)\exp(\beta H)\gamma(t)\ dt$$

Where $\lambda\uparrow+ (t)$ and $\gamma(t)$ are the Lagrange multiplier, $\rho\downarrow I,\ \ des$ is the target state.

* The gradient descent learning rule is given by:

$$\omega_{new} = \omega_{old} - \eta \frac{\partial L}{\partial \omega}$$

for each parameter ω={K, ϵ, ζ}, where:

$$\frac{\partial L}{\partial w} = \int_0^{t_f} \lambda^\dagger(t)(\frac{\partial \beta}{\partial t}[\rho_I, \frac{\partial H}{\partial w}] - \frac{i}{\hbar}[\rho_I, \frac{\partial H}{\partial w}] - \frac{i}{\hbar}\beta[[\rho_I, \frac{\partial H}{\partial w}], H])\gamma(t)dt$$

QMLM Aug. 2016

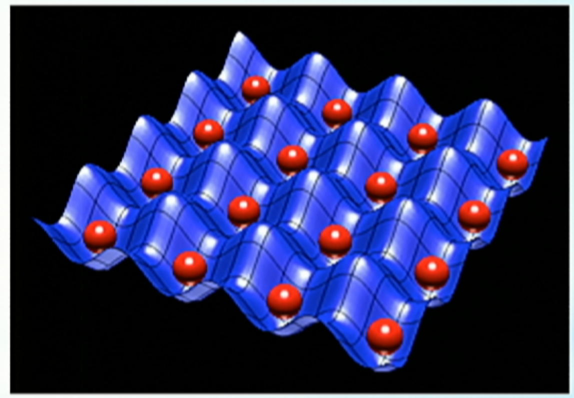
* Spatial Quantum Lattice Gas Network

Optical lattices use lasers to separate rubidium atoms (red) for use as information "bits" in neutral-atom quantum processors

Ψ for each lattice qubit Ψ =[↓ , ↑]

Spatial Schrodinger Equation:

$$i\hbar \frac{\partial \psi}{\partial t} = -K\sigma_x \nabla^2 \psi + [g|\phi|^2 - a]\psi$$



Behaves like a nonlinear quantum fluid

K(t) is a spatial tunneling between neighboring qubits

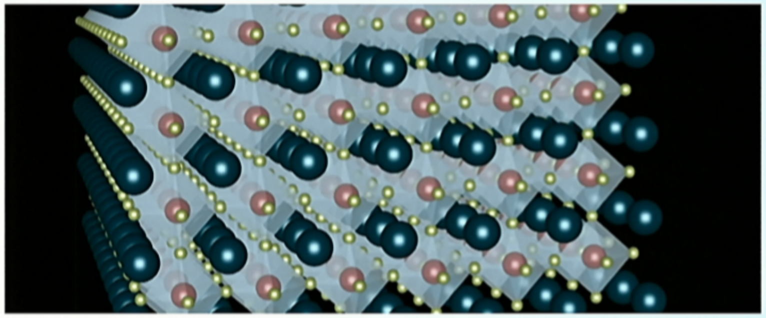
g(t) and; a(t) control a nonlinear bias term:

THESE ARE THE TRAINABLE "WEIGHTS" PARAMETERS

* Quantum Lattice Gas Network Learning

Cost Function

$$J = \frac{1}{2}[error]^2 + \int_0^{T_f}\int_G \lambda^\dagger \left[-K\sigma_x \nabla^2 \psi + [g|\phi|^2 - a]\psi - i\hbar \frac{\partial \psi}{\partial t} \right] dtdx$$



K(t) g(t), a(t)

ARE THE TRAINABLE "WEIGHTS" PARAMETERS

* Simulation: Quantum Lattice Gas Network

Integrate in Matlab Simulink over 40x50 2D grid lattice (4th order Runge-Kutta ODE4)

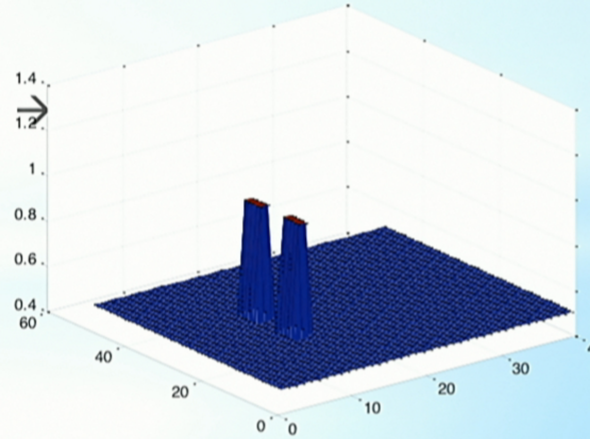
Total time = 200ns Δt=0.1ns Δx= Δy=√0.1

Initial Weight Parameters K=0.024, g=0.1, a=0.22

Inputs -> specify on an input region at (t=0)

2 Input Regions: Example of |↓> |↓> Input →

For a 2 input 1 output logic gate

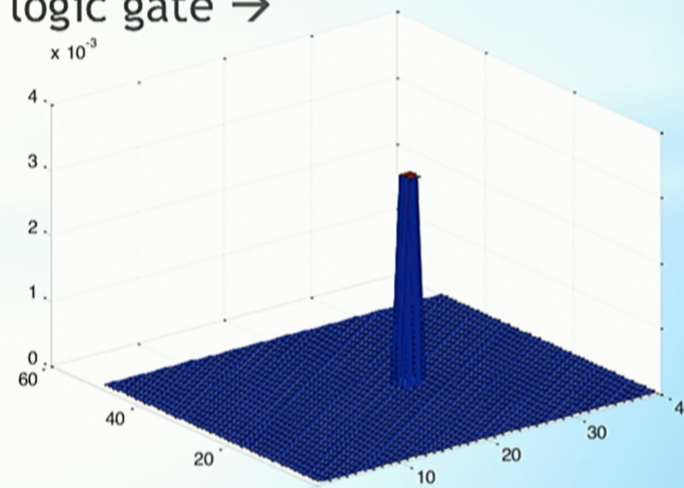


* Output: Quantum Lattice Gas Network

Outputs -> Measure of ψ on output region at $(t=T_f)$

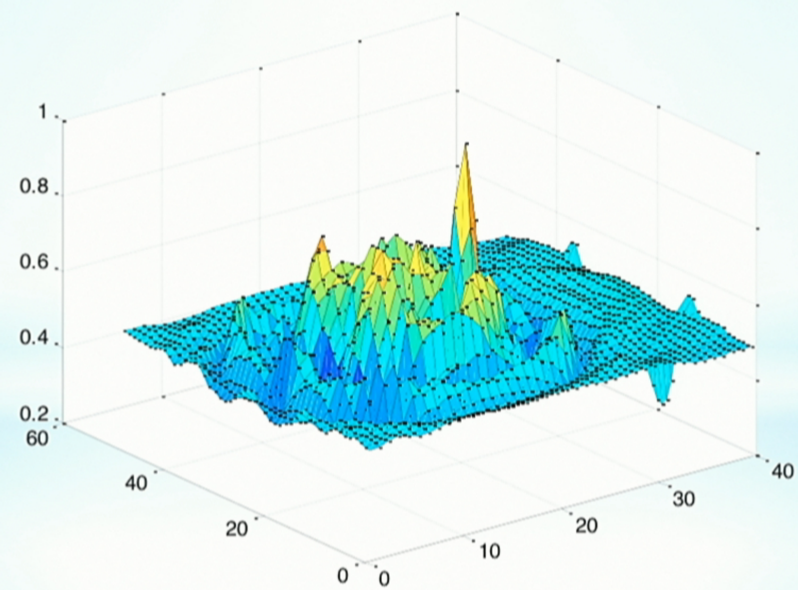
Targets -> target value for the Measured output

Output Region for 2 input 1 output logic gate →

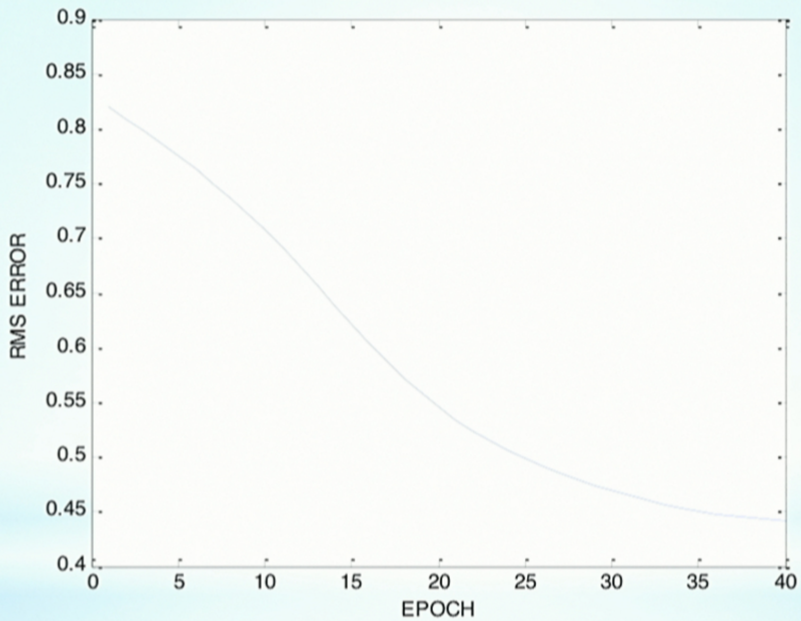


* Waves: Quantum Lattice Gas Network

| ↑ >   Probability   at Final Time for | ↑ > | ↑ >   Input



* RMS: Quantum Lattice Gas Network



- Need 3D grid to get Soliton behavior
- 2D Finite Difference grid too course but still computationally limited
- Need to port from PC to Cluster computer

* Algorithm construction
* Scaleup
* Noise
* Decoherence

* Single linear function
* Details of noise and decoherence robustness

*Conclusions

QMLM July 2016