Abstract: Machine learning is a rapidly growing field in computer science with applications in computer vision, voice recognition, medical diagnosis, spam filtering, search engines, etc. In this presentation, I will introduce a new machine learning approach based on quantum Boltzmann distribution of a transverse-field Ising Model. Due to the non-commutative nature of quantum mechanics, the training process of the Quantum Boltzmann Machine (QBM) can become nontrivial. I will show how to circumvent this problem by introducing bounds on the quantum probabilities. This allows training the QBM efficiently by sampling. I will then show examples of QBM training with and without the bound, using exact diagonalization, and compare the results with classical Boltzmann training. Finally, after a brief introduction to D-Wave quantum annealing processors, I will discuss the possibility of using such processors for QBM training and application.
Machine Learning Techniques

Model

Optimization

Search

Thanks to Ali Ghodsi (U of Waterloo)
Today's Tutorial
Deep reinforcement learning
Todd Sierens
4:00 pm, Bob room

Prep:
- Theano
- Lasagne
- SciKit: Neural Network
- numpy/matplotlib
Introduction to Machine Learning

Data → Model

Unseen data → Model
Introduction to Machine Learning

Data → Model

Unseen data → Model → 3

Copyright © 2016, D-Wave Systems Inc.
Boltzmann Machine

\( \mathbf{v} = [01100 \ldots 1] \)

Data

\[ \rightarrow P^\text{data}_v \]

Model

Variables \( \mathbf{v} \) Parameters \( \theta \)

\[ \rightarrow P_v(\theta) = \frac{e^{-E_v(\theta)}}{\sum_v e^{-E_v(\theta)}} \]

Boltzmann distribution \((\beta = 1)\)
Boltzmann Machine

Ising model:

\[ E_v(\theta) \rightarrow E_z = -\sum_a b_a z_a - \sum_{a,b} w_{ab} z_a z_b \]

\[ v_a = 0,1 \rightarrow z_a = \pm 1 \quad \text{spins} \]

\[ \theta \rightarrow b_a, w_{ab} \quad \text{parameters} \]
Adding Hidden Variables

$$E_z = -\sum_a b_a z_a - \sum_{a,b} w_{ab} z_a z_b$$

$$z_a = (z_v, z_i)$$

$$P_v = Z^{-1} \sum_h e^{-E_z}, \quad Z = \sum_z e^{-E_z}$$
Training a BM

Tune $\theta \in \{b_a, w_{ab}\}$ such that $P_v \approx P_v^{\text{data}}$

Maximize log-likelihood:

$$\sum_v P_v^{\text{data}} \log P_v$$

Or minimize:

$$\mathcal{L} = -\sum_v P_v^{\text{data}} \log P_v$$
Training a BM

**Tune** \( \theta \in \{b_a, w_{ab}\} \) such that \( P_v \approx P_v^{\text{data}} \)

**Maximize log-likelihood:**
\[
\sum_v P_v^{\text{data}} \log P_v
\]

**Or minimize:**
\[
\mathcal{L} = - \sum_v P_v^{\text{data}} \log P_v
\]

gradient descent technique

training rate
\[
\delta \theta = -\eta \partial_\theta \mathcal{L}
\]
Training Ising Hamiltonian Parameters

clamped

\[ \partial_\theta \mathcal{L} = \langle \partial_\theta E_z \rangle_v - \langle \partial_\theta E_z \rangle \]

unclamped

Average with clamped visibles

Unclamped average
Training Ising Hamiltonian Parameters

\[ E_z = - \sum_a b_a z_a - \sum_{a,b} w_{ab} z_a z_b \]

\[ \delta b_a = \eta \left( \langle z_a \rangle_v - \langle z_a \rangle \right) \]

Clamped average

Unclamped average

\[ \delta w_{ab} = \eta \left( \langle z_a z_b \rangle_v - \langle z_a z_b \rangle \right) \]
Question:

Is it possible to train a quantum Boltzmann machine?
Transverse Ising Hamiltonian

\[ H = - \sum_a \Gamma_a \sigma_a^x - \sum_a b_a \sigma_a^z - \sum_{a,b} w_{ab} \sigma_a^z \sigma_b^z \]

\[ \sigma_a^z \equiv \underbrace{I \otimes \cdots \otimes I}^{a-1} \otimes \sigma_z \otimes \underbrace{I \otimes \cdots \otimes I}_{N-a}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \]

\[ \sigma_a^x \equiv \underbrace{I \otimes \cdots \otimes I}^{a-1} \otimes \sigma_x \otimes \underbrace{I \otimes \cdots \otimes I}_{N-a}, \quad \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \]
Quantum Boltzmann Distribution

Marginal distribution: \( P_v = \text{Tr}[\Lambda_v \rho] \)

Density matrix: \( \rho = e^{-H}/\text{Tr}[e^{-H}] \)

\( \Lambda_v = |v\rangle \langle v| \otimes I_h \)

Projection operator  Identity matrix
Calculating the Gradient

\[ \mathcal{L} = -\sum_v P_v^{\text{data}} \log P_v = -\sum_v P_v^{\text{data}} \log \frac{\text{Tr}[\Lambda_v e^{-H}]}{\text{Tr}[e^{-H}]} \]

**Classical:** \([H, \partial_\theta H] = 0\)

\[ \Rightarrow \quad \partial_\theta e^{-H} = -e^{-H} \partial_\theta H \]

\[ \partial_\theta \mathcal{L} = \left( \langle \partial_\theta H \rangle_v - \langle \partial_\theta H \rangle \right) \]

**Clamped average**  **Unclamped average**
Calculating the Gradient

\[ \mathcal{L} = - \sum_v P_v^{\text{data}} \log P_v = - \sum_v P_v^{\text{data}} \log \frac{\text{Tr}[\Lambda_v e^{-H}]}{\text{Tr}[e^{-H}]} \]

**Quantum:** \([H, \partial_\theta H] \neq 0\)

\[ \Rightarrow \quad \partial_\theta e^{-H} \neq -e^{-H} \partial_\theta H \]

\[ \partial_\theta \mathcal{L} \neq \left( \langle \partial_\theta H \rangle_v - \langle \partial_\theta H \rangle \right) \]

- **Clamped average**
- **Unclamped average**
Two Useful Properties of Trace

\[ \partial_\theta e^{-H} \neq -e^{-H} \partial_\theta H \]
Two Useful Properties of Trace

\[ \text{Tr}[\partial_\theta e^{-H}] = -\text{Tr}[e^{-H} \partial_\theta H] \]
Two Useful Properties of Trace

\[ \text{Tr}[\partial_\theta e^{-H}] = -\text{Tr}[e^{-H} \partial_\theta H] \]

**Golden-Thompson inequality:**

For Hermitian matrices \( A \) and \( B \)

\[ \text{Tr}[e^A e^B] \geq \text{Tr}[e^{A+B}] \]
Finding lower bounds

\[ P_\nu = \frac{\text{Tr} [ \Lambda_\nu e^{-H} ]}{\text{Tr} [ e^{-H} ]} = \frac{\text{Tr} [ e^{-H} e^{\ln \Lambda_\nu} ]}{\text{Tr} [ e^{-H} ]} \geq \frac{\text{Tr} [ e^{-H+\ln \Lambda_\nu} ]}{\text{Tr} [ e^{-H} ]} \]

\[ P_\nu \geq \frac{\text{Tr} [ e^{-H_\nu} ]}{\text{Tr} [ e^{-H} ]} \quad H_\nu = H - \ln \Lambda_\nu \]

Visibles are clamped to data

Clamped Hamiltonian
Bound Optimization

\[ \mathcal{L} \lessapprox \tilde{\mathcal{L}} \equiv - \sum \eta_{\text{data}} \log \frac{\text{Tr}[e^{-H_{\nu}}]}{\text{Tr}[e^{-H}]} \]

\[ \partial_{\theta} \tilde{\mathcal{L}} = \left( \langle \partial_{\theta} H_{\nu} \rangle_{\nu} - \langle \partial_{\theta} H \rangle \right) \]

Clamped average \hspace{1cm} Unclamped average

\[ \delta b_{a} = \eta \left( \langle \sigma_{a}^{z} \rangle_{\nu} - \langle \sigma_{a}^{z} \rangle \right) \]

\[ \delta w_{ab} = \eta \left( \langle \sigma_{a}^{z} \sigma_{b}^{z} \rangle_{\nu} - \langle \sigma_{a}^{z} \sigma_{b}^{z} \rangle \right) \]
Exact Diagonalization Results

Amin, Andriyash, Rolfe, Kulchytskyy, Melko, arXiv:1601.02036
Exact Diagonalization Results

Amin, Andriyash, Rolfe, Kulchytskyy, Melko, arXiv:1601.02036

![Graph showing comparison between QBM, BM, and bQBM methods]

- **Bound gradient** \( \Delta = 2 \)
- **Classical BM**
- **Exact gradient** (\( \Delta \) is trained)
  \( \Delta_{\text{final}} = 2.5 \)
Question:

Can we use quantum annealing for training a quantum Boltzmann machine?
**D-Wave Quantum Annealer**

**D-Wave Hamiltonian:**

\[ H(t) = A(s)H_D + B(s)H_P \]

\[ H_D = -\sum_{i=1}^{N} \sigma_i^x \]

\[ H_P = \sum_{i=1}^{N} h_i \sigma_i^z + \sum_{i,j=1}^{N} J_{ij} \sigma_i^z \sigma_j^z \]

![Energy Functions Graph](graph.png)

- **A(s)**
- **B(s)**

Copyright © 2016, D-Wave Systems Inc.
Adiabatic Quantum Computation

\[ H(t) = (1-s)H_D + sH_P, \quad s = t/t_f \]
Adiabatic Quantum Computation

\[ H(t) = (1-s)H_D + sH_P, \quad s = t/t_f \]
Adiabatic Quantum Computation

\[ H(t) = (1-s)H_D + sH_P, \quad s = t/t_f \]

\[ t_f \sim \left(1/g_{\text{min}}\right)^2 \]
Adiabatic Quantum Computation

\[ H(t) = (1-s)H_D + sH_P, \quad s = t/t_f \]

\[ t_f \sim \left(\frac{1}{g_{\min}}\right)^2 \]
Thermal Noise

\[ H(t) = H_S(t) + H_B + H_{SB} \]

System  Bath  Interaction

Copyright© 2016, D-Wave Systems Inc.
Thermal Noise

\[ H(t) = H_S(t) + H_B + H_{SB} \]

- **System**
- **Bath**
- **Interaction**

- Energy levels
- \( k_B T \)
- \( P_0 \)

**Copyright© 2016, D-Wave Systems Inc.**
Thermal Noise

\[ H(t) = H_S(t) + H_B + H_{SB} \]

System  Bath  Interaction

\[ k_B T \]

0  \[ s \]  1

\[ P_0 \]
Thermal Noise

\[ H(t) = H_S(t) + H_B + H_{SB} \]

System  Bath  Interaction

Dynamical freeze-out
Equilibration During the Annealing

Amin, PRA 92, 052323 (2015)

Open quantum calculations of a 16 qubit random problem

Classical energies  Quantum energies

Equilibrium Probabilities

Copyright© 2016, D-Wave Systems Inc.
Example: 8-Qubit QBM

Fully connected (K8), fully visible

Logical graph:

Embedded graph:
Example: 8-Qubit QBM

Fully connected (K8), fully visible

Logical graph:

Embedded graph:
Logical Hamiltonian

**D-Wave Hamiltonian:** 
\[ H(t) = A(s)H_D + B(s)H_P \]

\[ H_D = -\sum_{i=1}^{N} \sigma_i^x \]

\[ H_P = \sum_{i=1}^{N} h_i \sigma_i^z + \sum_{i,j=1}^{N} J_{ij} \sigma_i^z \sigma_j^z \]

**Logical Hamiltonian (dimensionless):**

\[ H = -\sum_a \Gamma_a \sigma_a^x - \sum_a b_a \sigma_a^z - \sum_{a,b} w_{ab} \sigma_a^z \sigma_b^z \]

**Chain’s effective tunneling amplitude**

\[ b_a = \frac{4h_a B(s^*)}{k_B T} \]

\[ w_{ab} = \frac{2J_{ab} B(s^*)}{k_B T} \]

\[ \Gamma_a = \frac{\Delta_{eff}(s^*)}{2k_B T} \]
Training QBM

Amir Khoshaman
Training set: 4 Gaussian peaks

Amir Khoshaman
Training with Normal Annealing Schedule

- Classical BM
- Exact diagonalization ($\Gamma_a = 3.46$)
- DW with normal annealing

KLD vs. Epoch graph
Training with Normal Annealing Schedule

- Classical BM
- Exact diagonalization ($\Gamma_a = 3.46$)
- DW with normal annealing

KLD vs Epoch
Annealing with Ramp

D-Wave Hamiltonian: \( H(t) = A(s) \, H_D + B(s) \, H_P \)
Annealing with Ramp

D-Wave Hamiltonian: \[ H(t) = A(s) \, H_D + B(s) \, H_P \]
Annealing with Ramp

D-Wave Hamiltonian: \[ H(t) = A(s) \, H_D + B(s) \, H_P \]

Dickson et al., Nat. Commun. 4, 1903 (2013)
Correcting Distortions Due to the Ramp

If evolution during the ramp is local (1-2 qubits), a few sweeps of QMC can restore the distribution.

![Diagram showing projective readout, $s(t)$, $s^*$, Annealing with ramp (< 1µs), and normal annealing schedule with time $t$ and 20 µs along the x-axis.](image)
Correcting Distortions Due to the Ramp

If evolution during the ramp is local (1-2 qubits), a few sweeps of QMC can restore the distribution.
Quantum Monte Carlo Postprocessing

Distance from the exact quantum distribution at \( s^* = 0.3 \)

\( A(s^*) = 3.26 \text{ GHz}, \quad B(s^*) = 1.52 \text{ GHz}, \quad \Gamma_a(s^*) = 3.46 \)

![Graph showing KLD vs # of sweeps with QMC with random initialization and postprocessed DW samples]
Quantum Monte Carlo Postprocessing

Distance from the exact quantum distribution at $s^*=0.3$

$A(s^*)=3.26 \text{ GHz, } B(s^*)=1.52 \text{ GHz, } \Gamma_a(s^*)=3.46$

QMC with random initialization

Postprocessed DW samples
Training with Ramp + Postprocessing

- Classical BM
- Exact diagonalization
- Persistent QMC
- DW + postprocessing

KLD vs Epoch
Conclusions:

- A QBM can be trained by sampling
- A QBM may learn some distributions better than a classical BM
- A quantum annealer can provide samples for QBM training