Application of Quantum Annealing to Training of Deep Neural Networks

Steve Adachi, Ph.D.
Lockheed Martin

Workshop on Theory and Practice of Adiabatic Quantum Computers and Quantum Simulation
International Centre for Theoretical Physics, Trieste, Italy
22 Aug 2016
References

Our Paper


Related Work


Beyond Quantum Annealing / D-Wave

LM Research Partners in Quantum Information Science
(partial list)

University of Alberta
University of British Columbia
University of Southern California
Université de Sherbrooke
Dalhousie University
University of Maryland
University of Oxford
University College London
Griffith University
University of New South Wales
University of Sydney
University of Auckland

© 2016 Lockheed Martin Corporation. All rights reserved.
USC-Lockheed Martin Quantum Computation Center

• (May 2011) D-Wave Systems announced sale of first 128-qubit D-Wave One™ to Lockheed Martin.

• (Oct 2011) USC-Lockheed Martin Quantum Computing Center unveiled at USC Information Sciences Institute, Marina del Rey, CA.

• (Mar 2013) System upgraded to 512-qubit D-Wave Two™ (“Vesuvius”) chip.

• (Mar 2016) System upgraded to 1152-qubit D-Wave 2X™ (“Washington”) chip.
D-Wave hardware overview

**Qubit implementation**
- rf SQUID Flux Qubit
- Compound-Compound Josephson Junction

**Niobium on silicon**

**8-qubit unit cell**

**1152-qubit “Washington” chip**

**Magnetically shielded enclosure (10⁻⁹ Tesla)**

**Pulse tube dilution refrigerator**

Images © Copyright 2012-2016 D-Wave Systems Inc.
Questions

• Can a quantum annealing device be used to sample from a Boltzmann distribution?
• Can a quantum annealer assist in training a Restricted Boltzmann Machine?

Similarities

Quantum Annealer
(Ex. D-Wave Device)

$$\mathcal{H}_f = - \sum_i h_i \sigma_i^z - \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z$$

- Final states (in computational basis) are stochastic binary variables
- Quadratic energy functional
- Real device returns distribution of states (not 100% ground state) – can this be approximated as a Boltzmann distribution?

$$P \sim \frac{e^{-\beta_{eff} E'}}{Z'}$$

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

- Stochastic binary variables
- Quadratic energy functional
- Joint Boltzmann distribution

$$E(v, h) = - \sum_i b_i v_i - \sum_j c_j h_j - \sum_{ij} W_{ij} v_i h_j$$

© 2016 Lockheed Martin Corporation. All rights reserved.
References

**Our Paper**

**Related Work**

**Beyond Quantum Annealing / D-Wave**
Idea: How quantum sampling is applied to training of RBMs

• Restricted Boltzmann Machine model:

\[
E(v, h) = - \sum_i b_i v_i - \sum_j c_j h_j - \sum_{ij} W_{ij} v_i h_j
\]

Joint probability distribution

\[
P(v, h) = \frac{e^{-E}}{Z} \text{ where } Z = \sum_{v,h} e^{-E}
\]

• Weight updates are determined by the formula

\[
\Delta w_{ij} \propto \frac{\partial \log P}{\partial w_{ij}} = <v_i h_j>_{data} - <v_i h_j>_{model}
\]

• Second term is intractable; this has motivated approximate schemes such as Contrastive Divergence (CD):

\[
\text{“Contrastive Divergence” (CD-1)}: \quad \Delta w_{ij} \propto \langle H_1 V_1 \rangle - \langle H_0 V_0 \rangle
\]

• However, CD can take many iterations to converge (related to slow mixing of Gibbs sampling)

• We attempt to use quantum sampling to estimate the “intractable” term directly
  ▪ Quantum sampling has the potential to mix faster (e.g. due to tunneling)
Challenges using actual QA hardware for Boltzmann sampling

- **Limited physical connectivity between qubits**
  - Not a complete graph
  - Not a bipartite graph
  - “Chimera” graph (square lattice of $K_{4,4}$ unit cells)
  - Small number of faulty qubits

- **Parameter setting noise (aka Intrinsic Control Error (ICE))**
  - Multiple sources of error – some random, some systematic
  - Programmed coefficients $\neq$ actual coefficients
    - Approx. 4 bits of precision (D-Wave 2); higher on D-Wave 2X

- **Determination of $\beta_{eff}$ (equivalently, the effective temperature)**
  - We used a simple empirical rule of thumb based on RBM size
  - For a more systematic approach, see the talk by A. Perdomo-Ortiz
Mapping RBM bipartite graphs onto D-Wave chip

• Map each visible/hidden node to a chain of qubits:

  - Can map up to 32x32 RBM this way on a 504-qubit Vesuvius chip
  - How we handle faulty qubits:
    - Constrain RBM weights \( w_{ij} = 0 \) for missing couplers
    - Use voting on qubit chains to decide logical node values
      - Tunable voting threshold from 0.5 (majority) to 1.0 (consensus)

\[
E(v, h) = -b v - c h - vW h
\]

\[
P(v, h) = \frac{e^{-E}}{Z}
\]

\[
P(v, h) = \frac{e^{-\beta_{eff}E'}}{Z'} \quad \text{(ansatz)}
\]
Mitigating Control Errors – Gauge Transformations

• D-Wave is an analog device
  ▪ “Vesuvius” system has 4 bits precision
  ▪ Net of various sources of random & systematic error
  ▪ Example: “J-dependent h-offset”

• Ferromagnetic chains (added to do the mapping on previous slide) can exacerbate some of these effects

• Control errors can be partially mitigated by “gauge transformations”
  ▪ Re-define the meaning of problem variables by flipping a subset of the $S_i$
  ▪ Flipping $S_i$ induces a flip of the associated $h_i$ and $J_{ij}$
  ▪ Gauge transformation shown below (“basket weave”) is particularly helpful in mitigating J-dependent h-offset errors

RED qubits flipped
BLUE qubits unchanged
Test Case: “Coarse Grained” MNIST

MNIST data set ([http://yann.lecun.com/exdb/mnist](http://yann.lecun.com/exdb/mnist))

- Handwritten digits 0-9
- 60,000 training and 10,000 test set images with truth labels
- Each image consists of 784 greyscale pixels (28x28)

To fit the problem on Vesuvius, we “coarse-grained” the images:

- We discarded 2 pixels on each edge, leaving a 24x24 image
- We computed the average pixel value over each 4x4 block, resulting in a coarse-grained 6x6 image

- We discarded the 4 corners, resulting in 32 super-pixels
- A more challenging recognition problem than the real MNIST!
Results for CG-MNIST Data Set

100 post-training iterations

200 post-training iterations

400 post-training iterations

800 post-training iterations

© 2016 Lockheed Martin Corporation. All rights reserved.
Conclusions

• In this experiment, the quantum sampling-based training approach achieved higher accuracy than CD-1 training with fewer iterations of generative training

• More investigation needed to understand whether this is due to:
  - Better estimation of gradient → can this also be efficiently estimated classically?
  - Quantum effects

• Work in progress:
  - Larger quantum annealing devices (e.g. D-Wave 2X)
  - More sparsely connected RBMs

• Concept of using a quantum annealer for sampling/inference instead of optimization could lead to new applications for these devices
  - Also for circuit/gate based QC
Details of Quantum Sampling formulation

**Original RBM**

\[ E(v, h) = -bv - ch - vwh \]

**QUBO \((n+m) \times (n+m)\)**

\[
Q(x) = \frac{1}{\beta_{eff}} \begin{bmatrix}
  b & W \\
  0 & c
\end{bmatrix}
\]

**Ising model**

\[ E'(S) = -HS - SJS \]

**Embedded Ising model**

\[ E''(S) = -HS - SJS - SJ_{Fm}S \]

**Gauge transformed Ising model**

\[ E'''(\tilde{S}) = -\tilde{H}\tilde{S} - \tilde{S}\tilde{J}\tilde{S} - \tilde{S}\tilde{J}_{Fm}\tilde{S} \]

\[
\langle v_i h_j \rangle_{model} = \langle v_i h_j \rangle_{\beta_{eff}, E'} \cong \langle v_i h_j \rangle_{\beta_{eff}, E''}
\]

\[ \cong v_i h_j \]

assuming contributions from \(J_{Fm}\) terms are negligible

\[ \textit{NOTE: Don't use auto-scaling} \]

**Question:** With the D-Wave hardware noise and all the approximations we are making, this is going to be a noisy estimate of the log-likelihood gradient. But, could it be less noisy than Contrastive Divergence?
CG-MNIST experimental details

Modeled as a [ 32 32 32 10 ] network
Generated coarse-grained versions of all 60,000 training and 10,000 test images
→ “CG-MNIST” data set

**Generative training (pre-training)**

- Divided CG-MNIST training set into 5 sets of 12,000 images each

  - **Classical:** for N=1,2,3,…100
    - Trained a 32/32/32/10 DBN on each of the 5 12,000-image sets for N pre-training iterations
    - For each N and for each training set, we trained 20 networks (total 100 for each N)

  - **Quantum:** for N=1,2,3,…40, 50, 60, 70, 80, 90, 100
    - Trained a 32/32/32/10 DBN on each of the 5 12,000-image sets for N pre-training iterations
    - For each N and for each training set, we trained 1 network (total 5 for each N)
    - For each pre-training iteration we issued one solver call in each of 4 gauges w num_reads = 100 (total 400 samples), annealing_time=20, $\beta_{eff}=2$, voting threshold = 0.5, no mini-batching, learning rate = 0.1

**Discriminative training**

- Same for classical and quantum:
  - Applied truth labels and set last RBM layer coefficients using linear mapping
  - 10, 25, or 100 iterations of backpropagation using mini-batches of size 100