SOLVING SPARSE REPRESENTATIONS FOR OBJECT CLASSIFICATION

USING QUANTUM D-WAVE 2X MACHINE

Nga Nguyen, Amy Larson, Carleton Coffrin, John Perry, Gary Salazar, and Garrett Kenyon

Los Alamos National Laboratory & New Mexico Consortium

5th Neuro Inspired Computational Elements 2017, San Jose, Mar. 6-8, 2017
OUTLINE

A. SPARSE CODING ON A QUANTUM D-WAVE
B. CHOOSING DATASET
C. IMPLEMENTATION ON D-WAVE MACHINE
D. COMPARISON WITH CLASSICAL SOLVER
E. COMPRESSIVE SENSING
F. SUMMARY AND FUTURE WORK
Quantum D-Wave machine 2X: a quantum annealer

A. METHODOLOGY

Solving a sparse-coding (SC) problem

Objective function is of the form:

\[
E = \min_{\{\vec{a}, \phi\}} \left[ \frac{1}{2} |\vec{I} - \phi \vec{a}|^2 + \lambda \|\vec{a}\|_p \right].
\]

reconstruction error \hspace{1cm} L_p-sparseness penalty

Olshausen and Field, Nature 381, 607 (1996)
Rozell, Johnson, Baraniuk, and Olshausen, Neur. Comp. 20, 2526 (2008)
Quantum D-Wave machine 2X: a quantum annealer

A. METHODOLOGY

Solving a sparse-coding (SC) problem

Objective function is of the form:

\[ E = \min_{\{\mathbf{a}, \phi\}} \left[ \frac{1}{2} \left| \mathbf{I} - \phi \mathbf{a} \right|^2 + \lambda \| \mathbf{a} \|_p \right]. \]

- reconstruction error
- \(L_p\)-sparseness penalty

Olshausen and Field, Nature 381, 607 (1996)
Rozell, Johnson, Baraniuk, and Olshausen, Neur. Comp. 20, 2526 (2008)

- non-convex problem
- NP-hard class
Quantum D-Wave machine 2X: a quantum annealer

A. METHODOLOGY

Solving a sparse-coding (SC) problem

Objective function is of the form:

$$ E = \min_{\{\overrightarrow{a}, \phi\}} \left[ \frac{1}{2} |\overrightarrow{I} - \phi \overrightarrow{a}|^2 + \lambda ||\overrightarrow{a}||_p \right]. $$

reconstruction error \hspace{1cm} L_p\text{-sparseness penalty}

an example of SC reconstruction

Features (Receptive field)

\[ (\text{Receptive field}) \]

Activity

\[ (a_1, a_2, \ldots, a_n)^T = \]

courtesy of Xinhua Zhang
Quantum D-Wave machine 2X: a quantum annealer

A. METHODOLOGY

- mapping the sparse-coding problem onto a quantum unconstrained binary optimization (QUBO):

D-Wave Hamiltonian:

\[ H(h, Q, a) = \sum_i h_i a_i + \sum_{<i,j>} Q_{ij} a_i a_j \]

where \( a_i = \{0, 1\} \forall i \).
A. METHODOLOGY

mapping the sparse-coding problem onto a quantum unconstrained binary optimization (QUBO):

D-Wave Hamiltonian:

\[ H(h, Q, a) = \sum_i h_i a_i + \sum_{<i,j>} Q_{ij} a_i a_j \]

where \( a_i = \{0, 1\} \forall i \).

This mapping is achieved by the relations:

\[
\begin{align*}
    h &= -\phi^T \bar{I} + (\lambda + \frac{1}{2}), \\
    Q &= \frac{1}{2} \phi^T \phi.
\end{align*}
\]

analogous to L0-sparseness penalty [Nguyen and Kenyon, PMES-16 (2016)]
A. METHODOLOGY

4 “row” qubits
A. METHODOLOGY

4 “column” qubits
A. METHODOLOGY

unit cell
A. METHODOLOGY

intra-cell couplings
intra-cell couplings
A. METHODOLOGY

neighboring couplings
A. METHODOLOGY

neighboring couplings
B. DATASET

CIFAR-10

32x32

airplane

automobile

ship

truck

24x24

edge
detection
C. IMPLEMENTATION ON D-WAVE MACHINE

8 hand-designed features

“row”

“column”

orthogonality!

number of features $N_f = 8$
C. IMPLEMENTATION ON D-WAVE MACHINE

8 hand-designed features

"row"

= = =

= =

= =

orthogonality!

number of features $N_f = 8$

$\{\psi_i\}$

complete set: $\Phi = \Psi \ast \vec{I}$
8 hand-designed features

"row"

orthogonality!

number of features $N_f = 8$

$\{\psi_i\}$

complete set: $\Phi = \Psi \ast \overrightarrow{I}$

"column"
C. IMPLEMENTATION ON D-WAVE MACHINE

8 hand-designed features

"row"

orthogonality!

number of features $N_f = 8$

$\{\psi_i\}$

complete set: $\Phi = \Psi \times \vec{I}$

Desire: Randomly generated $N_f$:

$8 < N_f < 1152$
Building features

\[ N_f = 32 \]

\[ \cdots \]

\[ N_f = 1152 \]
• 24x24 patch images

8 and 32 features

1100 active qubits
3068 coupling strengths

overcomplete order:

$$2 = \frac{12 \times 12 \times 8}{24 \times 24 \times 1}$$

stride: 2, 4
• 24x24 patch images

32 and 1152 features

1100 active qubits
3068 coupling strengths

overcomplete order:

\[
2 = \frac{12 \times 12 \times 8}{24 \times 24 \times 1}
\]

stride: 4, 24
RESULTS

• 12x12 patch images

Classification task: SVM (liblinear)
1042 training/208 test images

<table>
<thead>
<tr>
<th>classes</th>
<th>air</th>
<th>auto</th>
<th>bird</th>
<th>cat</th>
<th>deer</th>
<th>dog</th>
<th>frog</th>
<th>horse</th>
<th>ship</th>
<th>truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>accur. (binary)</td>
<td>89.21%</td>
<td>93.38%</td>
<td>90.87%</td>
<td>94.71%</td>
<td>94.71%</td>
<td>88.94%</td>
<td>87.98%</td>
<td>89.9%</td>
<td>89.9%</td>
<td>85.58%</td>
</tr>
</tbody>
</table>

Nguyen and Kenyon, PMES-16 (2016)
So far, quantum computation (D-Wave 2X) has NOT outperformed its classical counterpart (GUROBI). Both are comparable.

We already made the problem hard. We need to make it harder.

How can we make the SC problem harder for both?
From **SC** perspective: more overcomplete, harder to solve…

**Meanwhile:** The full Chimera in D-Wave offers a certain set of *(nearest-neighbor)* connectivity…
From **SC** perspective: more overcomplete, harder to solve…

**Meanwhile:** The full Chimera in D-Wave offers a certain set of (**nearest-neighbor**) connectivity…

**EMBEDDING technique**
(c.f. D-Wave documents)
From SC perspective: more overcomplete, harder to solve…

Meanwhile: The full Chimera in D-Wave offers a certain set of (nearest-neighbor) connectivity…

**EMBEDDING technique**

(c.f. D-Wave documents)
From **SC** perspective: more overcomplete, harder to solve...

**Meanwhile**: The full Chimera in D-Wave offers a certain set of (*nearest-neighbor*) connectivity...

**EMBEDDING technique**

(c.f. D-Wave documents)
From **SC** perspective: more overcomplete, harder to solve…

**Meanwhile:** The full Chimera in D-Wave offers a certain set of *(nearest-neighbor)* connectivity…

**EMBEDDING technique**

- Employ all bipartite couplings
- Small number of nodes (qubits)
From **SC** perspective: more overcomplete, harder to solve...

**Meanwhile:** The full Chimera in D-Wave offers a certain set of (**nearest-neighbor**) connectivity...

**EMBEDDING technique**

- Employ all bipartite couplings
- Small number of nodes (qubits)

**In practice (D-Wave 2X):**
- Fully connected: 49 nodes
- Partially orthogonal: 74 nodes
- Feature optimization!
Feature Learning (in progress)

before…                        …after

5x5

5x5
D. COMPARISON WITH A CLASSICAL SOLVER

**STARTING TO SEE SOMETHING GOOD...**

No. of (random) Hamiltonians: **1**

<table>
<thead>
<tr>
<th>solver</th>
<th>GUROBI (best classical solver)</th>
<th>D-Wave 2X (ISING)</th>
</tr>
</thead>
<tbody>
<tr>
<td>problem</td>
<td>Energy</td>
<td>Energy</td>
</tr>
<tr>
<td>49 nodes:</td>
<td>Time</td>
<td>Time</td>
</tr>
<tr>
<td>fully connected</td>
<td>-29.99</td>
<td>-29.99</td>
</tr>
<tr>
<td></td>
<td>~ 480 seconds</td>
<td>few seconds</td>
</tr>
</tbody>
</table>

**D-Wave 2X**

-29.99

**GUROBI**

-29.99

**Energy**

-29.99

**Time**

few seconds
## Starting to See Something Good...

No. of (random) Hamiltonians: **1**

<table>
<thead>
<tr>
<th>Problem</th>
<th>GUROBI (best classical solver)</th>
<th>D-Wave 2X (ISING)</th>
</tr>
</thead>
</table>

D. Comparison with a Classical Solver

- **GUROBI** is the best classical solver.
- **D-Wave 2X** is the ISING solver.

- **Energy** values indicate the quality of the solution.
- **Time** values indicate the computation time.
COMPRESSIVE SENSING ON A QUANTUM MACHINE
WHY NOT?
COMPRESSIVE SENSING ON A QUANTUM MACHINE
WHY NOT?

COMPRESSIVE SENSING in a nutshell

\[
\begin{bmatrix}
y \\
\end{bmatrix}
= \begin{bmatrix}
\Phi \\
\end{bmatrix}
\begin{bmatrix}
x \\
\end{bmatrix}
\]

\(y\): measurement, sparse
\(x\): compressible signal
\(\Phi\): sensing matrix
COMPRESSIVE SENSING ON A QUANTUM MACHINE
WHY NOT?

COMPRESSIVE SENSING in a nutshell

\[
\begin{bmatrix}
  y \\
\end{bmatrix}
\begin{bmatrix}
  \Phi \\
\end{bmatrix}
\begin{bmatrix}
  x \\
\end{bmatrix}
\]

- \( y \): measurement, sparse
- \( x \): compressible signal
- \( \Phi \): sensing matrix

\( L_0 \): NP-hard
\( L_1 \): mostly-used

wide applications particularly in image processing (X-ray, CT, ...), sampling, etc (c.f. Candes, Baraniuk, Compressive Sensing)
COMPRESSIVE SENSING ON A QUANTUM MACHINE

Abel transform algorithm

Radiograph Experiment

Obtaining radial density profile $f(r)$ of a cylindrically symmetric object

Charge coupled device camera

X-rays

Object

Scintillator

cross-section
COMPRESSIVE SENSING ON A QUANTUM MACHINE

Abel transform algorithm

Projected radial density of a cylindrically symmetric object:

\( h(y) \) projected, measurable
COMPRESSIVE SENSING ON A QUANTUM MACHINE

Abel transform algorithm

projected radial density of a cylindrically symmetric object:
\( h(y) \)

projected, measurable

Task: Recovering the density profile of the cylindrical object \( f(r) \):
COMPRESSIVE SENSING ON A QUANTUM MACHINE

Abel transform algorithm

projected radial density of a cylindrically symmetric object:
\[ h(y) \]
projected, measurable

Task: Recovering the density profile of the cylindrical object \( f(r) \):
\[ h(y) = 2 \int_{y}^{R} \frac{f(r)rdr}{\sqrt{(r^2 - y^2)}} \]

How?:
one method:
to obtain a slice of the original object
Abel inversion
\[ h(y) = A \ast f(r) \]
COMPRESSIVE SENSING ON A QUANTUM MACHINE

Abel transform algorithm + D-Wave QUBO

Our method is applicable to e.g. X-ray images as sparse inputs

\[ h = A f \]

We obtain:

\[ A_{ij} = 2 \sum_{i | r_i = y_j} \frac{r_i \Delta_{ij}}{\sqrt{r_i^2 - y_j^2}} \]
COMPRESSIVE SENSING ON A QUANTUM MACHINE

Abel transform algorithm + D-Wave QUBO

Our method is applicable to e.g. X-ray images as sparse inputs

\[ h = A f \]

We obtain:

\[ A_{ij} = 2 \sum_{i | r_i = y_j} \frac{r_i \Delta_{ij}}{\sqrt{r_i^2 - y_j^2}} \]

Next: to choose (learn) domain basis:

\[ f(r) = \sum_n s_n F_n(r) \]

\( S_n \) are sparse Fourier coefficients.
COMPRESSIVE SENSING ON A QUANTUM MACHINE

Abel transform algorithm + D-Wave QUBO

Our method is applicable to e.g. X-ray images as sparse inputs

\[ h = AF \]

We obtain: \[ A_{ij} = 2 \sum_{i|r_i=R, \ i|y_i=y_j} \frac{r_i \Delta_{ij}}{\sqrt{r_i^2 - y_j^2}} \]

then:

\[ h = AFs \]

Next: to choose (learn) domain basis:

\[ f(r) = \sum_n s_n F_n(r) \]

\( s_n \) are sparse Fourier coefficients.

Ising input
D-Wave Ising pulls out ALL non-zero Fourier coefficients (frequencies)
**COMPRESSIVE SENSING ON A QUANTUM MACHINE**

**Results**

- **D-Wave Ising** pulls out **ALL** non-zero Fourier coefficients (frequencies)

---

**Graph:**
- Signal $f(r)$ recovered from its original using quantum D-Wave Ising (LANL) machine.

**Table:**

<table>
<thead>
<tr>
<th>Problem</th>
<th>Solver</th>
<th>GUROBI</th>
<th>D-Wave 2X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compressive sensing on radiograph images</td>
<td>energy</td>
<td>time</td>
<td>energy</td>
</tr>
<tr>
<td>~ -6.84</td>
<td>cutoff 9 hrs</td>
<td>-67.673</td>
<td>few sec</td>
</tr>
</tbody>
</table>
E. SUMMARY

- First demonstration of sparse coding using quantum computer
- Benchmark results on standard image classification task
- Mapping of visual features to D-Wave Chimera
- Compare D-Wave performance with GUROBI
- Compressive sensing on Ising for density profile detection where D-Wave significantly outperforms GUROBI
E. FUTURE WORK

- optimize features
- add colors
- TrueNorth comparison
- hierarchy model
- compressive sensing on real images

THANK YOU FOR YOUR ATTENTION!