On the speed-up of adiabatic quantum computers by anomaly detection on IP traffic datasets

Quantum Computing and High Performance Computing
CINECA – 15/12/2022 – Casalecchio di Reno (BO)

Lorenzo Moro¹,³, Enrico Prati²,³

¹ – POLIMI
² – UNIMI
³ – CNR-IFN
Restricted Boltzmann Machines

Applications:
- Recommendation systems,
- Network Anomaly Detection,
- Fraud detection,
- Quantum tomography,
- …

Generative neural network models

- Lessons from the Netflix Prize Challenge
  - Robert M. Bell and Yehuda Koren
  - 2007

- Improved traffic detection with support vector machine based on restricted Boltzmann machine
  - Jun Yang1, Jianglei Deng1, Shujuan Li1, Yongle Hao1
  - 30 Dec, 2015

- Abnormal Traffic Pattern Detection in Real-Time Financial Transactions
  - Saam Rastatter, Travis Moe, Amitava Gangopadhyay and Alfred Whaer
  - 13 Mar, 2019

- Restricted Boltzmann machines in quantum physics
  - Roger G. McIver, Giuseppe Carleo, Juan Carrasquilla and Ignacio Cirac
  - Nature Physics 15, 887-892(2019) | Cite this article
  - June 2019
Restricted Boltzmann Machine

Hidden units (latent space)

Visible units (input/output)

Binary variables:
- \( h_i = \{-1, 1\} \)
- \( v_i = \{-1, 1\} \)

Parameters (weights and biases)

\[ W \quad a \quad b \]
Restricted Boltzmann Machine

At each state is associated with an energy \( E(s) \)

\[
E(S) = E(v,h) = - \sum_{i \in \text{visibles}} a_i v_i - \sum_{j \in \text{hidden}} b_j h_j - \sum_{i,j} v_i W_{ij} h_j
\]

Hidden units (latent space)

\( h_1 \), \( h_2 \), \( h_3 \)

Visible units (input/output)

\( v_1 \), \( v_2 \), \( v_3 \)

Binary variables:

\( h_j = \{-1,1\} \)

\( V_i = \{-1,1\} \)

Parameters (weights and biases)

\( W \)  
\( a \)  
\( b \)

The joint probability \( P(v,h) \) is a Boltzmann distribution

\[
P(v,h) = \frac{e^{-E(v,h)}}{\sum_{v,h} e^{-E(v,h)}}
\]
Restricted Boltzmann Machine

At each state is associated with an energy $E(s)$

$$E(s) = E(v, h) = - \sum_{i \in \text{visibles}} a_i v_i - \sum_{j \in \text{hidden}} b_j h_j - \sum_{i,j} v_i W_{ij} h_j$$

The joint probability $P(v,h)$ is a Boltzmann distribution

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

To sample from a RBM you can:

$$P(h_j = 1 | v) = \sigma(b_j + \sum_j v_j W_{ij})$$

$$P(v_i = 1 | h) = \sigma(a_i + \sum h_i W_{ij})$$

$$\sigma(x) = \frac{e^x}{1 + e^x}$$

The goal is to train weights and biases

Binary variables:
- $h_j = \{-1, 1\}$
- $V_i = \{-1, 1\}$

Parameters (weights and biases)
- $W_{ij}$
- $a_i$ and $b_j$
Training a RBM

RBM is trained by maximizing the likelihood of training data

\[ \mathcal{L}(W, a, b) = \sum_{v \in \text{data}} \log P(v) \]

and performing gradient ascent

\[ \nabla_{ij} \mathcal{L}(W, a, b) = \sum_{v \in \text{data}} \left( \sum_{H} v_i h_j e^{-E(v, H)} \right) \left( \sum_{H} e^{-E(v, H)} \right) - N \left( \sum_{V, H} v_i h_j e^{-E(V, H)} \right) \]

\[ \frac{Z}{N} \]
Training a RBM

RBM is trained by maximizing the likelihood of training data

$$l l(W, a, b) = \sum_{v \in data} \log P(v)$$

and performing gradient ascent

$$\nabla_{ij} ll(W, a, b) = \sum_{v \in data} \sum_{H} v_{i} h_{j} e^{-E(v, H)} - N \sum_{V, H} v_{i} h_{j} e^{-E(V, H)} \frac{Z}{Z}$$

Positive phase: Easy to compute

Negative phase: Very hard to compute (Partition function Z)
Training a RBM

RBM is trained by maximizing the likelihood of training data

$$ll(W,a,b) = \sum_{v \in \text{data}} \log P(v)$$

and performing gradient ascent

$$\nabla_{ij} ll(W,a,b) = \sum_{v \in \text{data}} \frac{\sum_{H} v_i h_j e^{-E(v,H)}}{Z} - N \sum_{v,H} v_i h_j e^{-E(v,H)}$$

Positive phase: Easy to compute

Negative phase: Very hard to compute (Partition function $Z$)

Contrastive Divergence doesn't follow the gradient of any function

Contrastive Divergence (CD-K)

Still, it works fine...


Sutskever, Ilya, and Tijmen Tieleman. "On the convergence properties of contrastive divergence."

ON THE SPEED-UP OF ADIABATIC QUANTUM COMPUTERS BY ANOMALY DETECTION ON IP TRAFFIC DATASETS

L. Moro @ PoliMI - CNR
Training a RBM

RBM is trained by maximizing the likelihood of training data

$$\mathbb{I}(W,a,b) = \sum_{v \in \text{data}} \log P(v)$$

and performing gradient ascent

$$\nabla_{ij} \mathbb{I}(W,a,b) = \sum_{v \in \text{data}} \sum_{H} v_i h_j e^{-E(v,H)} - N \sum_{V,H} v_i h_j e^{-E(v,H)}$$

Positive phase: Easy to compute

Negative phase: Very hard compute (Partition function $Z$)

Contrastive Divergence doesn’t follow the gradient of any function

Still, it works fine...

\[ \text{RBMs classical computational cost is high!!!} \]
Training a QRBM

RBM is trained by maximizing the likelihood of training data

$$ll(W, a, b) = \sum_{v \in \text{data}} \log P(v)$$

and performing gradient ascent

$$\nabla_{ij} ll(W, a, b) = \sum_{v \in \text{data}} \frac{\sum_{H} vihj e^{-E(v, H)}}{\sum_{H} e^{-E(v, H)}} - N \frac{\sum_{V, H} vihj e^{-E(V, H)}}{Z}$$

Negative phase: 
Very hard compute

Positive phase: 
Easy to compute

Or we can embed the RBM on the QPU and sample the states!!

$$H(t) = -F(t) (\sum_{i} B_{i} \sigma_{i}^{f} \sum_{i,j} J_{ij} \sigma_{i}^{f} \sigma_{j}^{f}) - I(t) \sum_{i} \sigma_{i}^{f}$$

Weights

Bias ($a \ e \ b$)

At each annealing cycle you produce samples with probability

$$P(v, h) = \frac{e^{-E(v, h) \over T_{\text{eff}}}}{\sum_{v, h} e^{-E(v, h) \over T_{\text{eff}}}}$$

$T_{\text{eff}}$ can be set

We can estimate the negative phase at the cost of a single quantum operation!
Quantum advantage

What does it mean “quantum advantage” for QRBMs?
Quantum advantage

What does it mean “quantum advantage” for QRBM’s?

Increase in performance metrics

(accuracy, reconstruction error, precision, compression ratio)

We are focusing on a classification problem, so we measured the accuracy and F1 score

\[
F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}
\]

\[
\text{accuracy} = \frac{TP + TN}{TP + FP + FN + TN}
\]
Quantum advantage

What does it mean “quantum advantage” for QRBMs?

Increase in performance metrics

Speed-up the model

(accuracy, reconstruction error, precision, compression ratio)

We are focusing on a classification problem, so we measured the accuracy and F1 score

\[
F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}
\]

\[
\text{accuracy} = \frac{TP + TN}{TP + FP + FN + TN}
\]

Could mean reducing:

- Increase in performance metrics
- Speed-up the model

The computational complexity

- The computational time

Or both

ON THE SPEED-UP OF ADIABATIC QUANTUM COMPUTERS BY ANOMALY DETECTION ON IP TRAFFIC DATASETS

L. Moro @ PoliMI - CNR
The data: real-world cybersecurity datasets

This database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment.

Network dataset consists of seven weeks of raw TCP/IP dump files of various attack was used against a local-area network (LAN) simulating a typical U.S. Air Force LAN.

Attacks fall into four main categories:

- **DoS**: denial-of-service, e.g. syn flood;
- **R2L**: unauthorized access from a remote machine, e.g. guessing password;
- **U2R**: unauthorized access to local superuser (root) privileges, e.g., various “buffer overflow” probing: surveillance and other probing, e.g., port scanning.

The dataset is generated in a systematic manner. It contains detailed descriptions of intrusions and abstract distribution models for applications, protocols, or lower level network entities.

**Goal:** classify attacks from normal activities

- **DoS and DDoS**: denial-of-service, e.g. hulk, goldeneye ecc...;
- **Bruteforce attack**: e.g. guessing password;
- **Web attack**: In-house selenium framework; Damn Vulnerable Web App;
- **Botnet attack**: Zeus, which is a Trojan horse malware package for Windows

CSE-CIC-IDS2018 on AWS
The only difference between a RBM and QRBM is the training procedure. In particular how the negative phase is evaluated. However:

- The contrastive divergence (CD-k) procedure works surprisingly well if k>>1
- The quantum sampling is better, but needs an ideal quantum annealer (no environment coupling, complete superposition, H implemented exactly, no errors)

The only difference between a RBM and QRBM is the training procedure. In particular how the negative phase is evaluated. However:

- The contrastive divergence (CD-k) procedure works surprisingly well if $k \gg 1$
- The quantum sampling is better, but needs an ideal quantum annealer (no environment coupling, complete superposition, $H$ implemented exactly, no errors)

We achieved the same performance by employing 1 CD-k step and extracting 10 quantum samples from the QPU!!
Computational complexity

Contrastive divergence procedure scales linearly in the number of RBM units $N$ (visible + hidden) and in the number of CD-K steps.

Computational complexity

The computational complexity represents the number of operations to perform to complete the computation.

Contrastive divergence procedure scales linearly in the number of RBM units $N$ (visible + hidden) and in the number of CD-K steps:

$$\text{RBM complexity} \sim O(N \cdot K)$$

Ideally, quantum sampling scales linearly in the number of samples extracted $(S)$ but doesn’t depend on the QRBM size:

$$\text{QRBM complexity} \sim O(N \cdot S)$$

However, AQC are physical devices restrained by technological and engineering constraints.

Classical and quantum machines have the same computational complexity.

Computational Times

Classic times depend on the number of contrastive divergence steps (K) and number of qubits involved.

**Single core CPU times**

128 cores CPU times

L. Moro, E. Prati, "On the speed-up of adiabatic quantum computers by anomaly detection of IP traffic", in preparation
Computational Times

Quantum times depend on the number of sample extracted and number of qubits involved.

Classic times depend on the number of contrastive divergence steps (K) and number of qubits involved.

Single core CPU times

Classic times depend on the number of contrastive divergence steps (K) and number of qubits involved.

128 cores CPU times

Advantage 4.1 QPU times

Quantum times depend on the number of sample extracted and number of qubits involved.
Quantum times depend on the number of samples extracted and number of qubits involved.

Classic times depend on the number of contrastive divergence steps (K) and number of qubits involved.

**Single core CPU times**

Advantage 4.1 QPU times

**128 cores CPU times**

- CSE-CIC-IDS2018
- NSL-KDD

Quantum times depend on the number of sample extracted and number of qubits involved.

**ON THE SPEED-UP OF ADIABATIC QUANTUM COMPUTERS BY ANOMALY DETECTION ON IP TRAFFIC DATASETS**

L. Moro, E. Prati, "On the speed-up of adiabatic quantum computers by anomaly detection of IP traffic", in preparation
The quantum speed-up is problem-dependent. It could emerge only for sufficiently large RBM or for tasks that require a high number of Gibbs steps.
The number of CD steps during inference hugely affect the performance of the model. During the training it is ok having a noisy gradient ascent step.

To maximize the performance we need to perform CD-10 and CD-100 on the NSL-KDD and CSE-CIC_IDS2018 datasets, respectively.

### Inference Times

<table>
<thead>
<tr>
<th>Dataset</th>
<th>k</th>
<th>Accuracy</th>
<th>F1</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>TN</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSL-KDD</td>
<td>1</td>
<td>0.906±0.005</td>
<td>0.901±0.005</td>
<td>0.909±0.009</td>
<td>0.091±0.009</td>
<td>0.096±0.005</td>
<td>0.903±0.005</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.935±0.002</td>
<td>0.932±0.002</td>
<td>0.936±0.003</td>
<td>0.064±0.003</td>
<td>0.065±0.003</td>
<td>0.935±0.003</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.937±0.002</td>
<td>0.934±0.002</td>
<td>0.939±0.004</td>
<td>0.061±0.004</td>
<td>0.064±0.001</td>
<td>0.935±0.001</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>0.938±0.002</td>
<td>0.935±0.002</td>
<td>0.979±0.002</td>
<td>0.020±0.003</td>
<td>0.064±0.002</td>
<td>0.936±0.002</td>
</tr>
<tr>
<td>CSE-CIC_IDS2018</td>
<td>1</td>
<td>0.800±0.004</td>
<td>0.805±0.005</td>
<td>0.833±0.003</td>
<td>0.166±0.004</td>
<td>0.234±0.006</td>
<td>0.766±0.006</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.897±0.001</td>
<td>0.903±0.001</td>
<td>0.907±0.002</td>
<td>0.093±0.002</td>
<td>0.113±0.002</td>
<td>0.887±0.002</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>0.915±0.001</td>
<td>0.921±0.001</td>
<td>0.922±0.003</td>
<td>0.078±0.003</td>
<td>0.092±0.001</td>
<td>0.907±0.001</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>0.924±0.001</td>
<td>0.929±0.001</td>
<td>0.932±0.001</td>
<td>0.068±0.002</td>
<td>0.084±0.001</td>
<td>0.916±0.001</td>
</tr>
</tbody>
</table>
The number of CD steps during inference hugely affect the performance of the model.

During the training it is ok having a noisy gradient ascent step.

We detected a quantum speedup in the query time.
However...

RBMcs are inferred locally

QRBMcs are inferred on the cloud

Data

Cloud latency

CPU

ON THE SPEED-UP OF ADIABATIC QUANTUM COMPUTERS BY ANOMALY DETECTION ON IP TRAFFIC DATASETS

L. Moro @ PoliMI - CNR
However...

The CPU can process data in batches.

The QPU have to process on data at the time.

RBM are inferred locally.

QRBM are inferred on the cloud.
CONCLUSION

We trained RBM and QRBM on two real-world cybersecurity datasets

QRBMs don't present a computational complexity advantage on current quantum hardware (connectivity problem)

QRBMs haven't shown better performance than RBMs on the task (no increase in accuracy/F1)

The quantum speed-up is problem dependent

RBMs training is not faster on quantum computer (contrastive divergence works well)

We measured a quantum speed-up! (QRBMs have a shorter inference time)

However:
1) cloud latency prevents small models from a quantum advantage
2) QPU cannot process batch of data
3) can we lower the quantum computational complexity improving qubit connectivity?

Next week on Arxiv:
L. Moro, E. Prati, "On the speed-up of adiabatic quantum computers by anomaly detection of IP traffic", in preparation

Contact:
enrico.prati@unimi.it

This work was partially founded by Vista Technology
CONCLUSION

We trained RBM and QRBM on two real-world cybersecurity datasets

QRBM doesn't present a computational complexity advantage on current quantum hardware (connectivity problem)
QRBM haven't shown better performance than RBM on the task (no increase in accuracy/F1)

The quantum speed-up is problem dependent
RBMs training is not faster on quantum computer (contrastive divergence works well)

We measured a quantum speed-up! (QRBM have a shorter inference time)

Next week on Arxiv:
L. Moro, E. Prati, "On the speed-up of adiabatic quantum computers by anomaly detection of IP traffic", in preparation

However:
1) cloud latency prevents small models from a quantum advantage
2) QPU cannot process batch of data
3) can we lower the quantum computational complexity improving qubit connectivity?

THANK YOU

Contact:
enrico.prati@unimi.it

This work was partially founded by Vista Technology