POLITECNICO
MILANO 1863

Consiglio Nazionale
delle Ricerche

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)

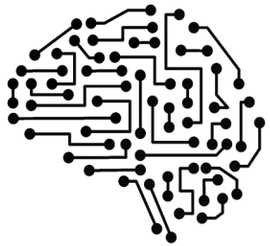
Lorenzo Moro[1,3], Enrico Prati[2,3]

1 – POLIMI
2 – UNIMI
3 – CNR-IFN

Restricted Boltzamann Machines

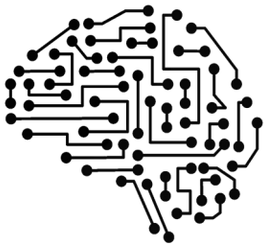
Restricted Boltzmann Machines Generative neural network models

Dataset



RBM

Corrupted input



Reconstructed output

RBMs applications:

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

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 Yang[1] · Jiangdong Deng[1] · Shujuan Li[1] · Yongle Hao[1] 30 Dec, 2015

Abnormal Traffic Pattern Detection in Real-Time Financial Transactions 13 Mar, 2019

Sean Rastatter, Travis Moe, Amitava Gangopadhyay and Alfred Weaver

Perspective | Published: 24 June 2019

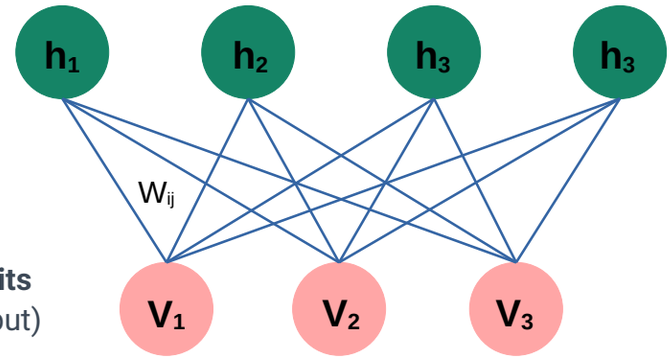
Restricted Boltzmann machines in quantum physics

Roger G. Melko ✉, Giuseppe Carleo, Juan Carrasquilla & J. Ignacio Cirac June 2019

Nature Physics **15**, 887–892(2019) | Cite this article

Restricted Boltzmann Machine

Hidden units
(latent space)



Visible units
(input/output)

Binary variables:

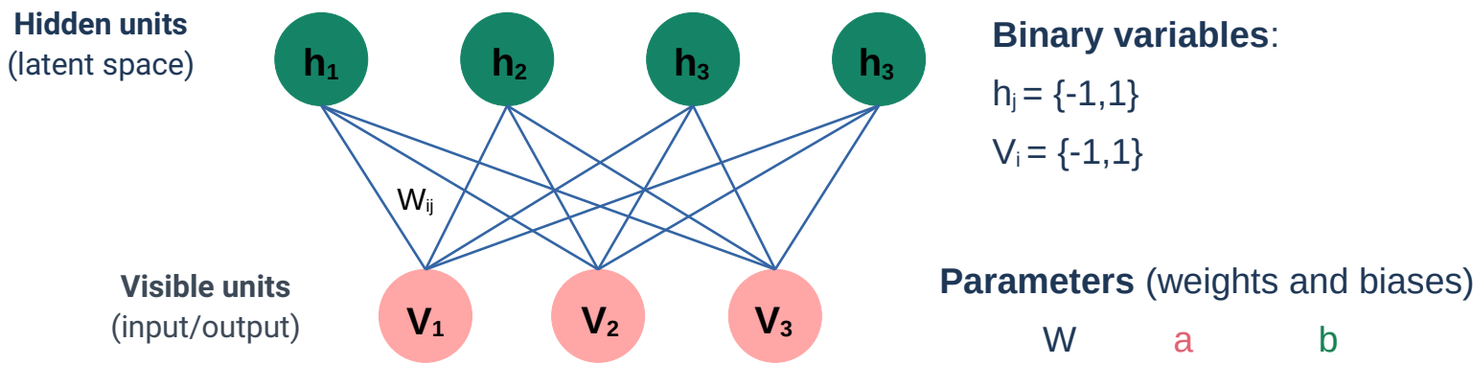
$h_j = \{-1,1\}$

$V_i = \{-1,1\}$

Parameters (weights and biases)

W a b

Restricted Boltzmann Machine



Binary variables:

$h_j = \{-1,1\}$

$V_i = \{-1,1\}$

Parameters (weights and biases)

W a b

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

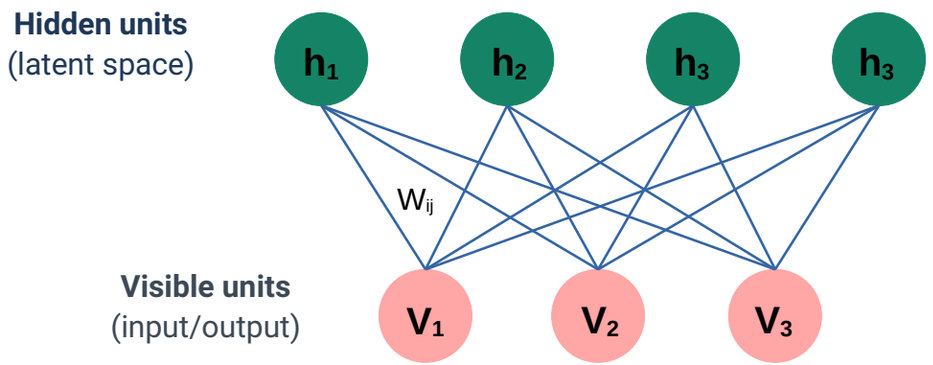
$$E(S) = E(v,h) = -\sum_{i \in visibles} a_i v_i - \sum_{j \in hidden} b_j h_j - \sum_{i,j} v_i W_{ij} h_j$$

Visible bias — Hidden bias — Weights

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



Binary variables:

$h_j = \{-1,1\}$

$V_i = \{-1,1\}$

Parameters (weights and biases)

W a b

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

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

Visible bias — Hidden bias — Weights

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

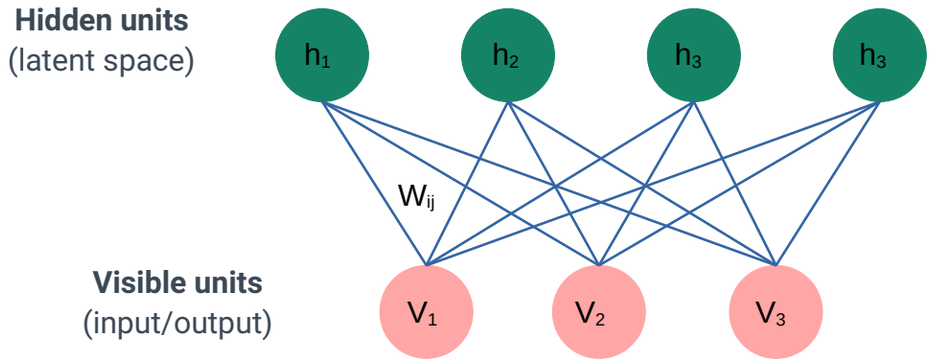
$$P(v,h) = \frac{e^{-E(v,h)}}{\sum_{v,h} e^{-E(v,h)}}$$

To sample from a RBM you can:

$$P(h_j = 1|v) = \sigma\left(b_j + \sum_j v_j W_{ij}\right) \qquad P(v_i = 1|h) = \sigma\left(a_i + \sum_i h_i W_{ij}\right) \qquad \sigma(x) = \frac{e^x}{1+e^x}$$

The goal is to train weights and biases

Training a RBM



RBM is trained by maximizing the likelihood of training data

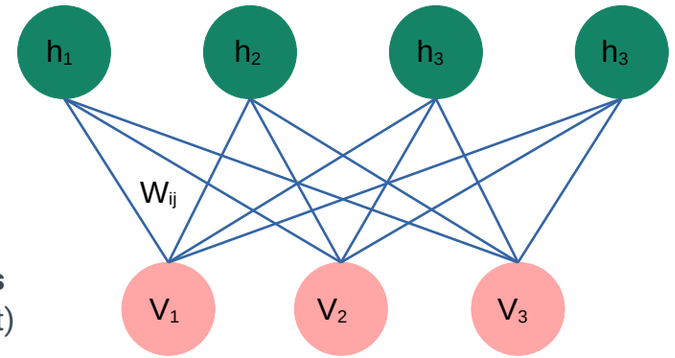
$$ll(W,a,b)=\sum_{v\in data}\log P(v)$$

and performing gradient ascent

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

Training a RBM

Hidden units
(latent space)



Visible units
(input/output)

RBM is trained by maximizing the likelihood of training data

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

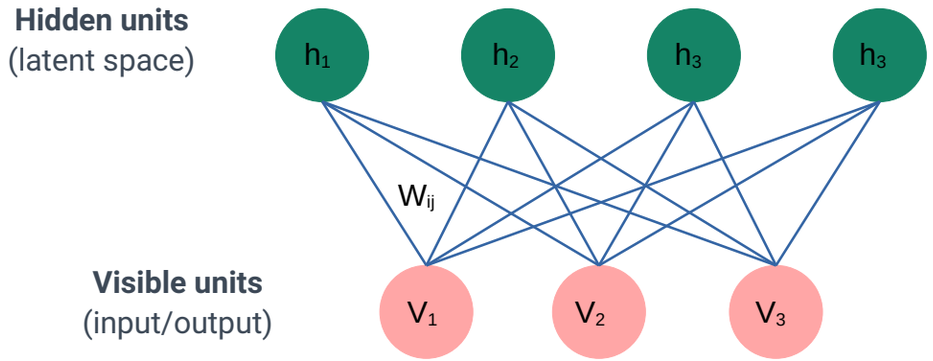
and performing gradient ascent

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

Positive phase:
Easy to compute

Negative phase:
Very hard compute
(Partition function Z)

Training a RBM



RBM is trained by maximizing the likelihood of training data

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

and performing gradient ascent

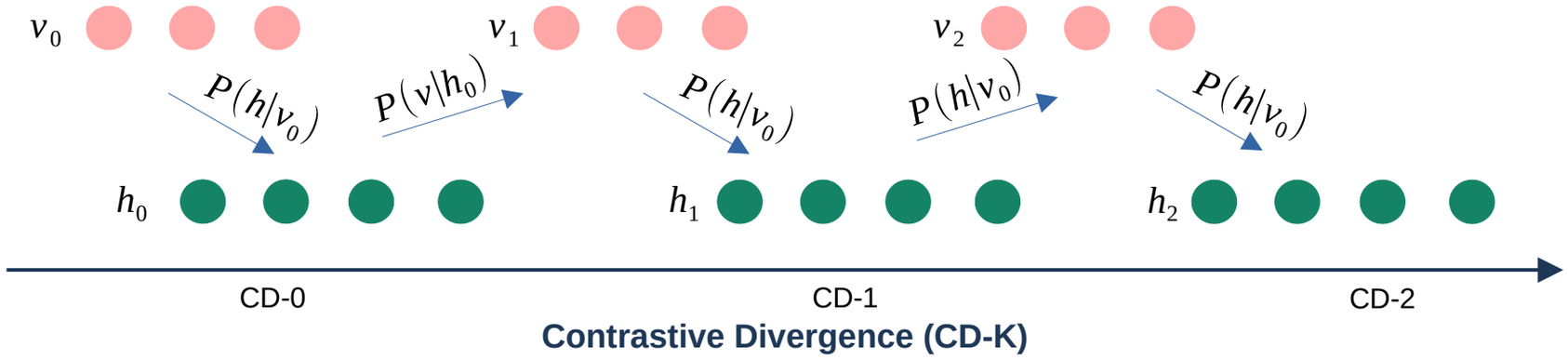
$$\nabla_{ij} ll(W,a,b)=\sum_{v\in data}\frac{\sum_H v_i h_j e^{-E(v,H)}}{\sum_H e^{-E(v,H)}}-N\frac{\sum_{V,H} v_i h_j e^{-E(V,H)}}{Z}$$

Positive phase:
Easy to compute

Negative phase:
Very hard compute
(Partition function Z)

Classical Training (CD-k)

Ackley, David H., Geoffrey E. Hinton, and Terrence J. Sejnowski. "A learning algorithm for Boltzmann machines."



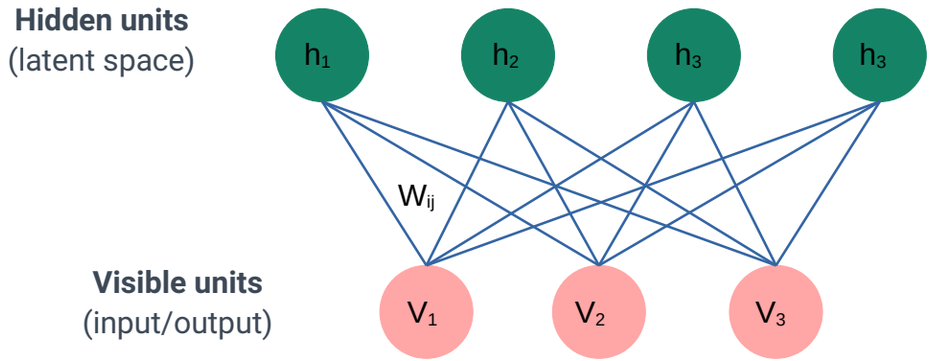
Contrastive Divergence (CD-K)

Contrastive Divergence doesn't follow the gradient of any function

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

Still, it works fine...

Training a RBM



RBM is trained by maximizing the likelihood of training data

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

and performing gradient ascent

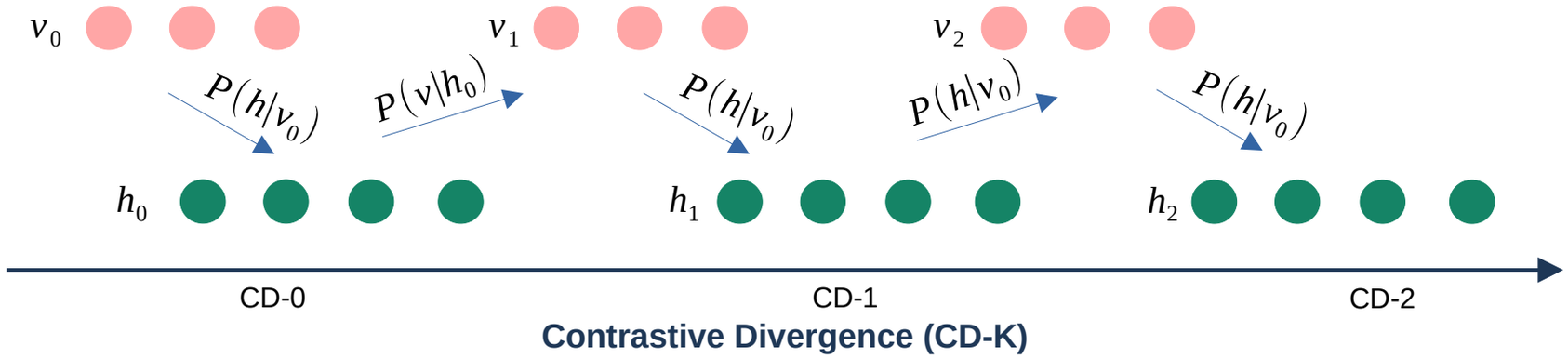
$$\nabla_{ij}\,ll(W,a,b)=\sum_{v\in data}\frac{\sum_{H}v_i h_j e^{-E(v,H)}}{\sum_{H}e^{-E(v,H)}}-N\frac{\sum_{V,H}v_i h_j e^{-E(V,H)}}{Z}$$

Positive phase:
Easy to compute

Negative phase:
Very hard compute
(Partition function Z)

Classical Training (CD-k)

Ackley, David H., Geoffrey E. Hinton, and Terrence J. Sejnowski. "A learning algorithm for Boltzmann machines."



Contrastive Divergence (CD-K)

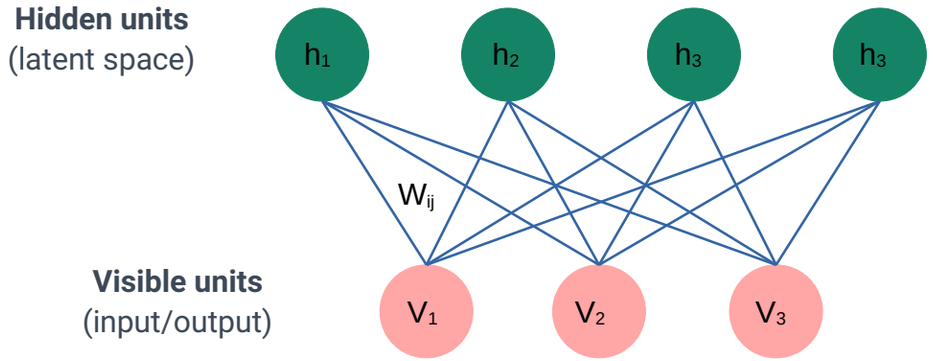
Contrastive Divergence doesn't follow the gradient of any function

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

Still, it works fine...

RBMs classical computational cost is high!!!

Training a QRBM



RBM is trained by maximizing the likelihood of training data

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

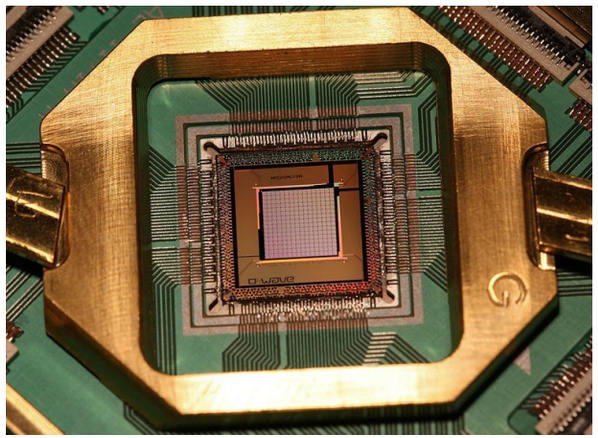
and performing gradient ascent

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

Positive phase:
Easy to compute

Negative phase:
Very hard compute
(Partition function Z)

Quantum Training



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

$$H(t)=-F(t)\left(\sum_i B_i\,\sigma_i^z\sum_{i,j}J_{ij}\,\sigma_i^z\,\sigma_j^z\right)-I(t)\sum_i\sigma_i^x$$

Weights — Bias (a e b)

At each annealing cycle you produce samples with probability

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

→ T_{eff} can be set

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

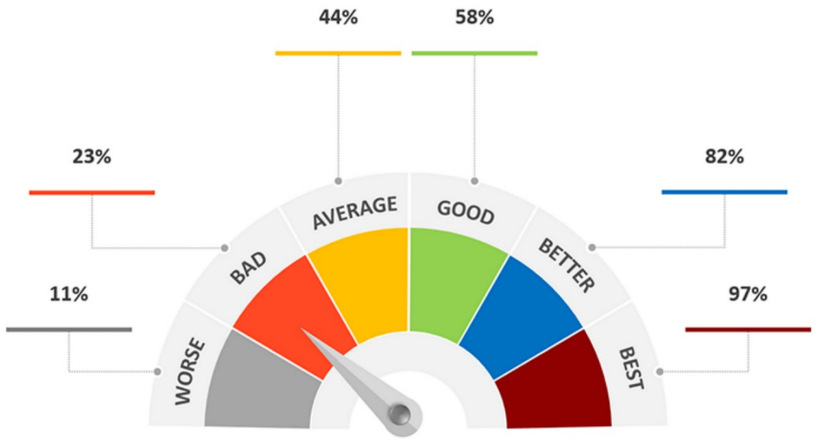
Quantum advantage

What does it mean “quantum advantage” for QRBMs?

Quantum advantage

What does it mean “quantum advantage” for QRBMs?

Increase in performance metrics



(accuracy, reconstruction error, precision, compression ratio)

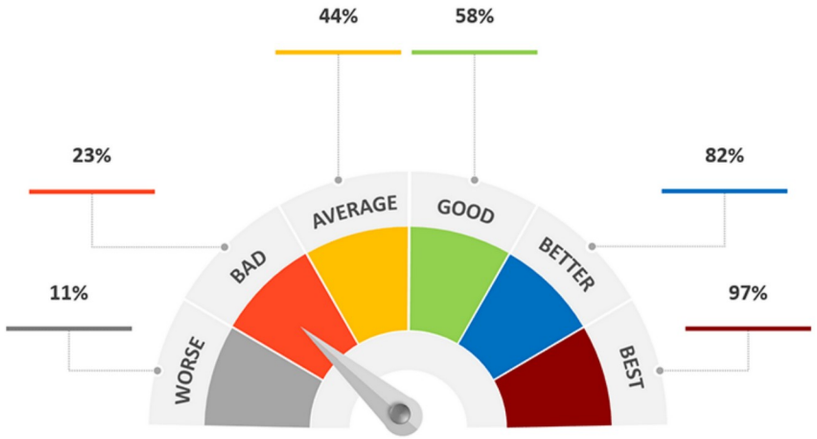
We are focusing on a classification problem, so we measured the accuracy and F1 score

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \qquad accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Quantum advantage

What does it mean “quantum advantage” for QRBMs?

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

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \qquad accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Speed-up the model

Could mean reducing:

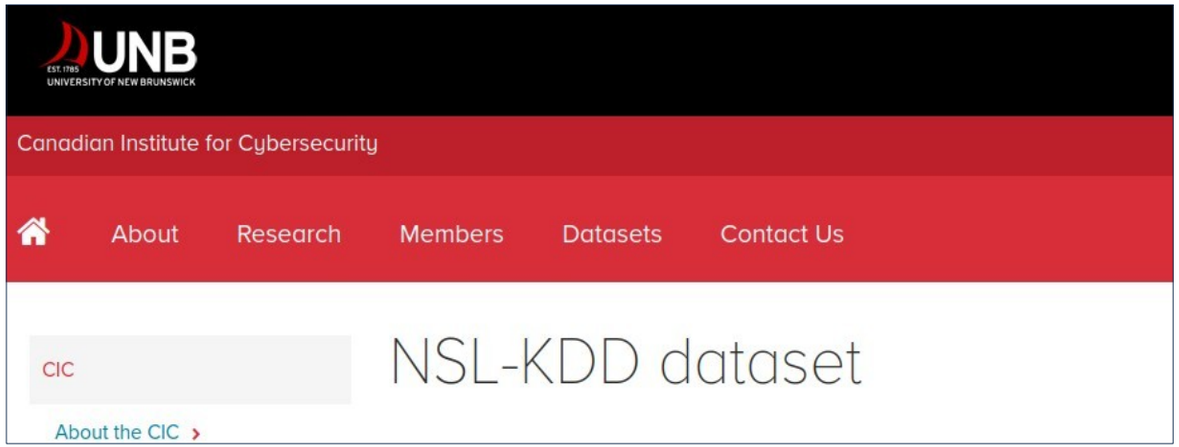
The computational complexity

The computational time

- **Training time**
- **Query time**

Or both

The data: real-world cybersecurity datasets



GOAL: classify attacks from normal activities

L.Moro, E.Prati, "A quantum annealing restricted boltzmann machine for cybersecurity detection systems", in preparation

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

DARPA → (Feature extracted) → **KDD99** → (Duplicates removed, Size reduced) → **NSL-KDD**

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:

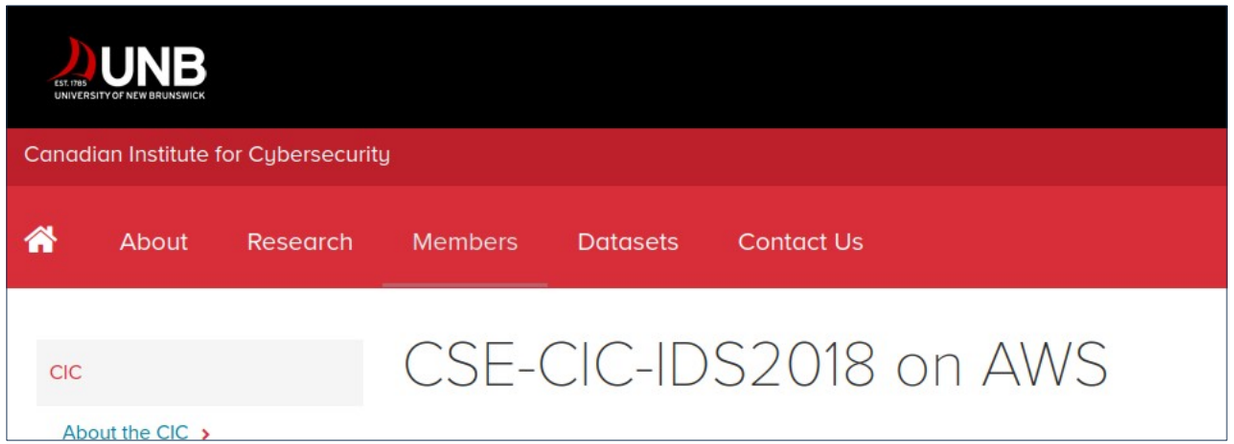
DoS: denial-of-service, e.g. syn flood;
R2L: unauthorized access from a remote machine, e.g. guessing password;
U2R: unauthorized access to local superuser (root) privileges, e.g., various ``buffer overflow''
probing: surveillance and other probing, e.g., port scanning.

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

CSE-CIC-IDS2018

Feature extracted, but no pre-processed

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



Increase in performance

Average performance

	NSL-KDD		CSE-CIC-IDS2018	
	Accuracy	**False Positive Rate**	**Accuracy**	**False Positive Rate**
RBM	94%	2%	92%	7%
QRBM	94%	5%	85%	16%

Visible units: 85 and 156

Hidden units: 30 and 90

Training epochs: 2000

RBM **CD-K: 300**

QRBM **# samples: 100**

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

QRBMs have roughly the same performace of RBMs

However, usally classical RBM performs better (quantum hardware issue?)

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)

Carreira-Perpinan, Miguel A., and Geoffrey Hinton. "On contrastive divergence learning." International workshop on artificial intelligence and statistics. PMLR, 2005.

Increase in performance

Average performance

	NSL-KDD		CSE-CIC-IDS2018	
	Accuracy	**False Positive Rate**	**Accuracy**	**False Positive Rate**
RBM	94%	2%	92%	7%
QRBM	94%	5%	85%	16%

Visible units: 85 and 156

Hidden units: 30 and 90

Training epochs: 2000

RBM **CD-K: 300**

QRBM **# samples: 100**

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

QRBMs have roughly the same performace of RBMs

However, usally classical RBM performs better (quantum hardware issue?)

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)

Carreira-Perpinan, Miguel A., and Geoffrey Hinton. "On contrastive divergence learning." International workshop on artificial intelligence and statistics. PMLR, 2005.

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

Computational complexity

The computational complexity represent the the number of operation to perform to complete the computation.

RBM

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

RBM complexity ~ O(N · K)

Rocutto, L., Destri, C. & Prati, E. Quantum semantic learning by reverse annealing of an adiabatic quantum computer. Adv. Quantum Technol. (2021).

Computational complexity

The computational complexity represent the the number of operation to perform to complete the computation.

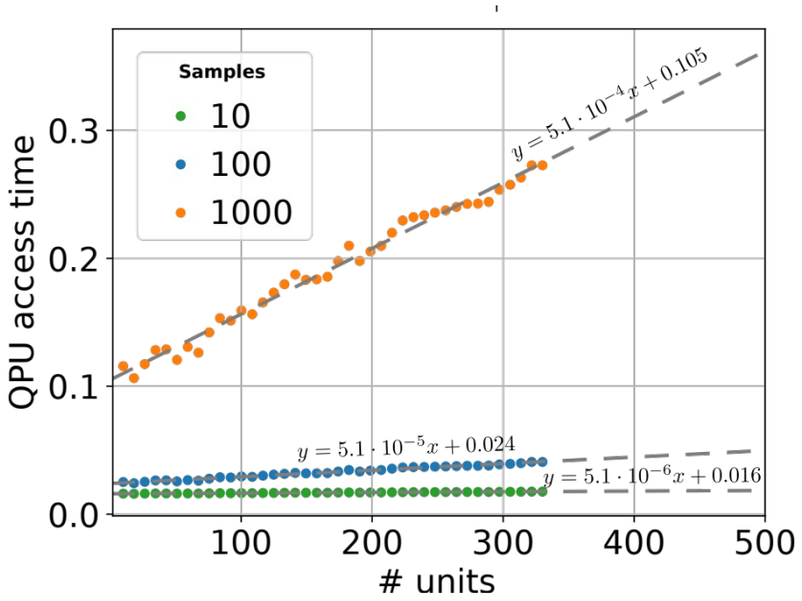
RBM

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

RBM complexity ~ O(N · K)

Rocutto, L., Destri, C. & Prati, E. Quantum semantic learning by reverse annealing of an adiabatic quantum computer. Adv. Quantum Technol. (2021).

QRBM



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

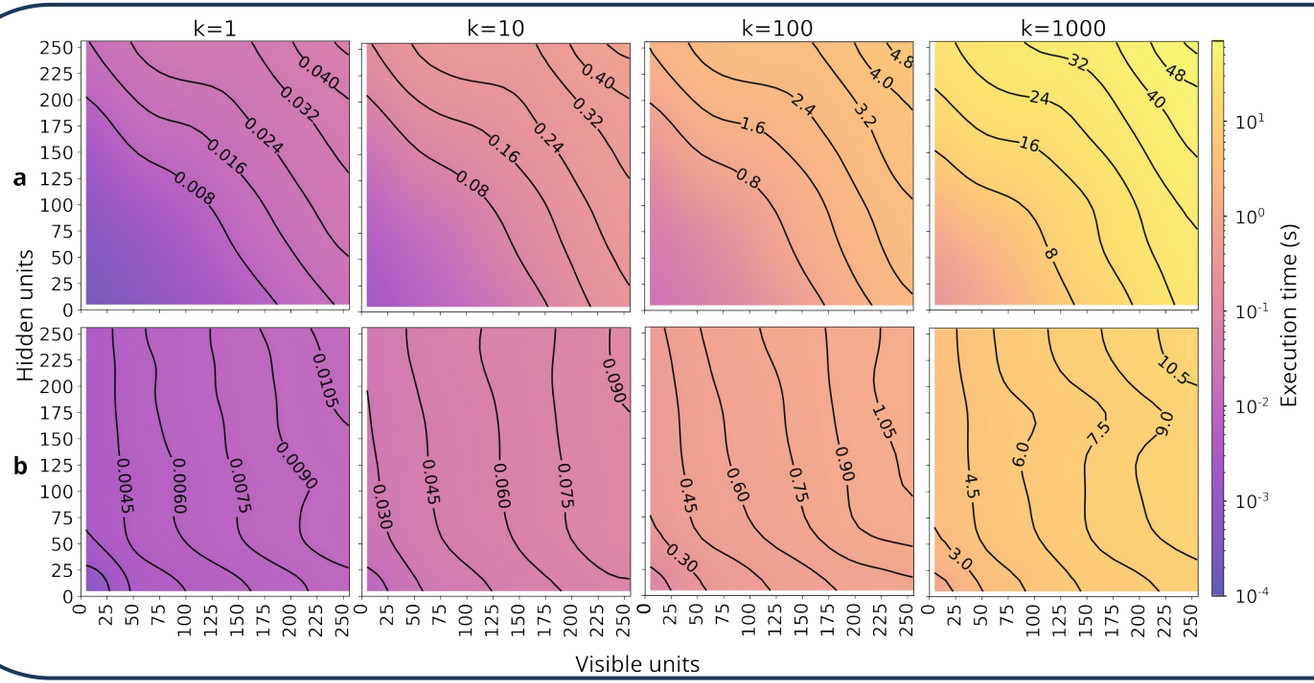
However AQC are physical devices restrained by technological and engineering constraint

QRBM complexity ~ O(N · S)

Classical and quantum machines have the same computational complexity

Computational Times

RBM



Single core CPU times

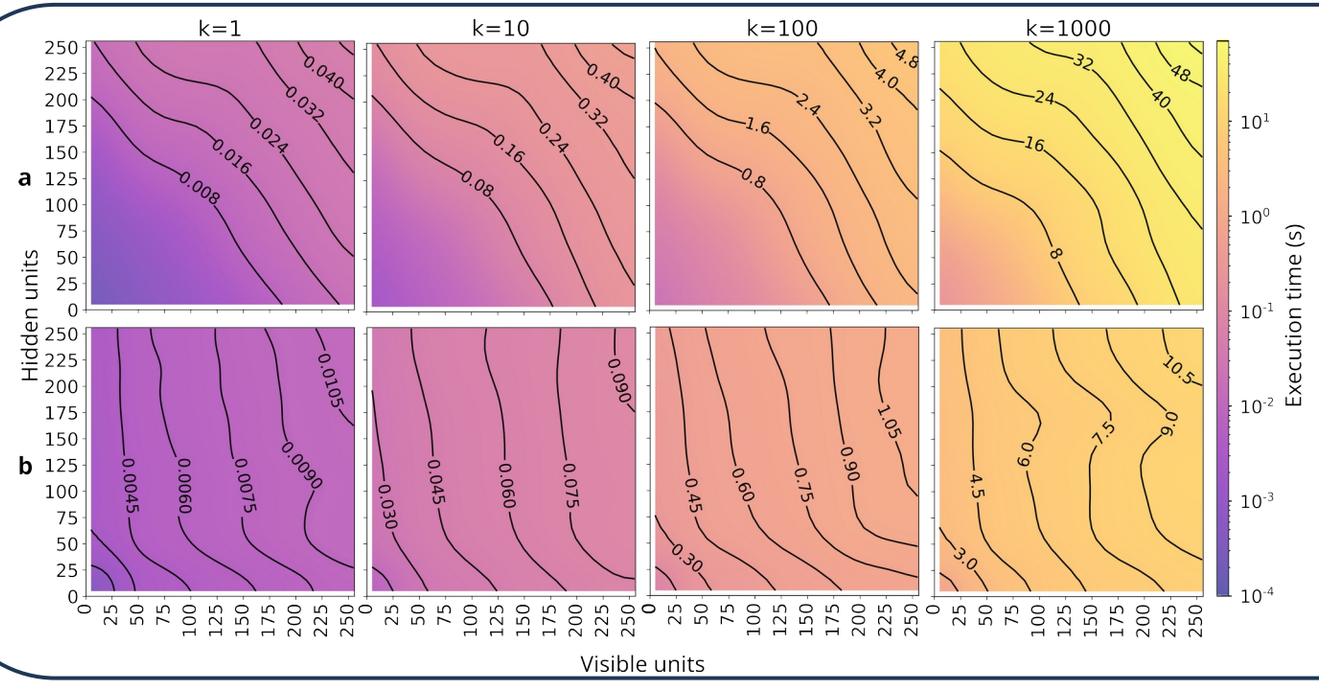
L.Moro, E.Prati, “On the speed-up of adiabatic quantum computers by anomaly detection of IP traffic”, in preparation

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

128 cores CPU times

Computational Times

RBM



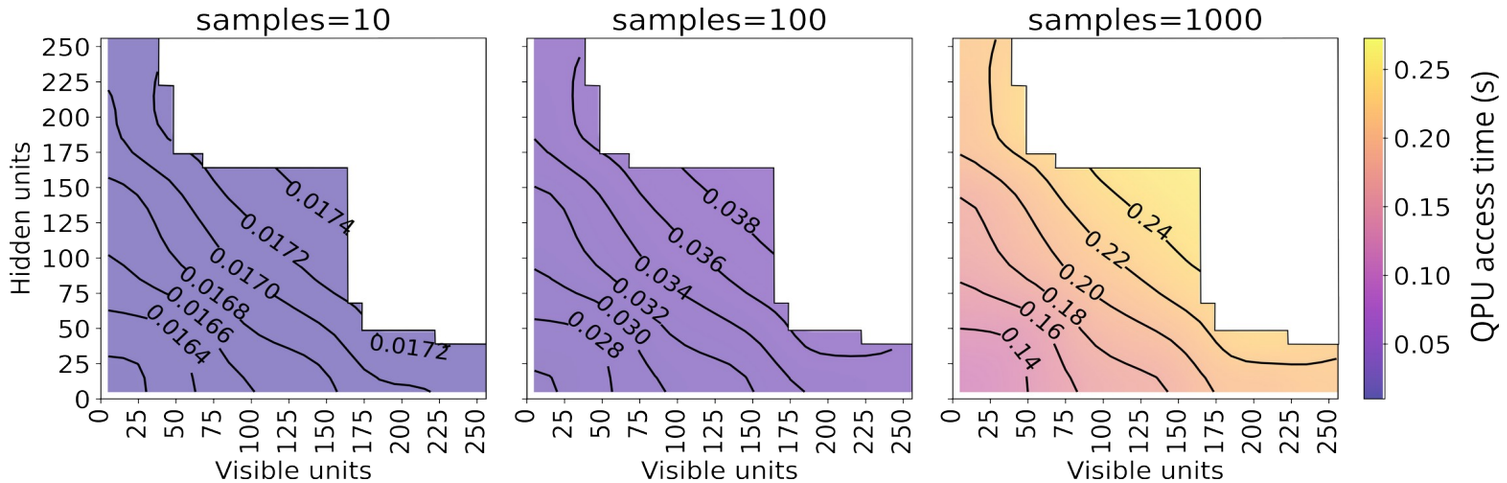
Single core CPU times

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

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

128 cores CPU times

QRBM

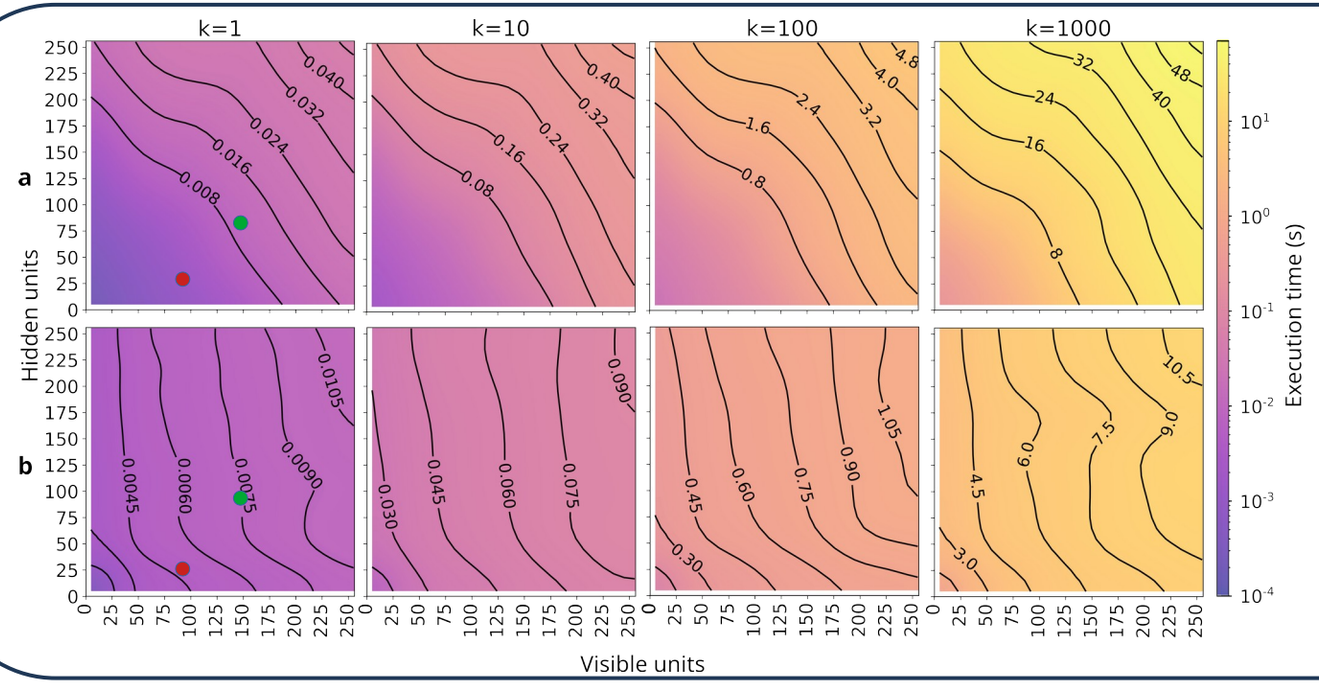


Advantage 4.1 QPU times

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

Computational Times

RBM



Single core CPU times

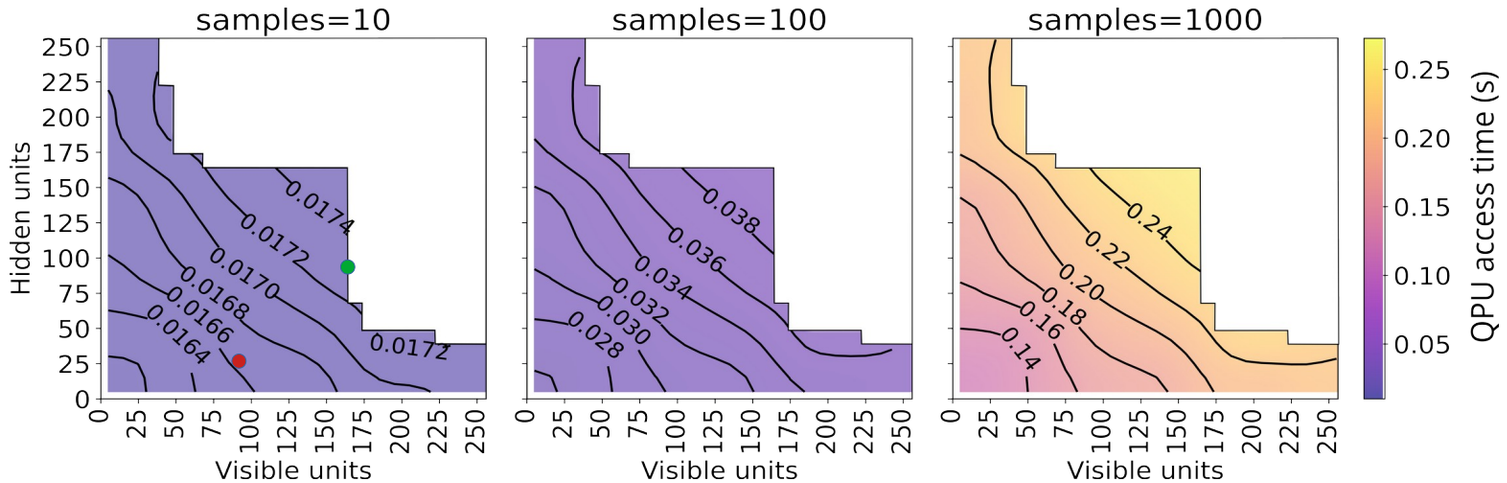
L.Moro, E.Prati, “On the speed-up of adiabatic quantum computers by anomaly detection of IP traffic”, in preparation

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

128 cores CPU times

- CSE-CIC-IDS2018
- NSL-KDD

QRBM

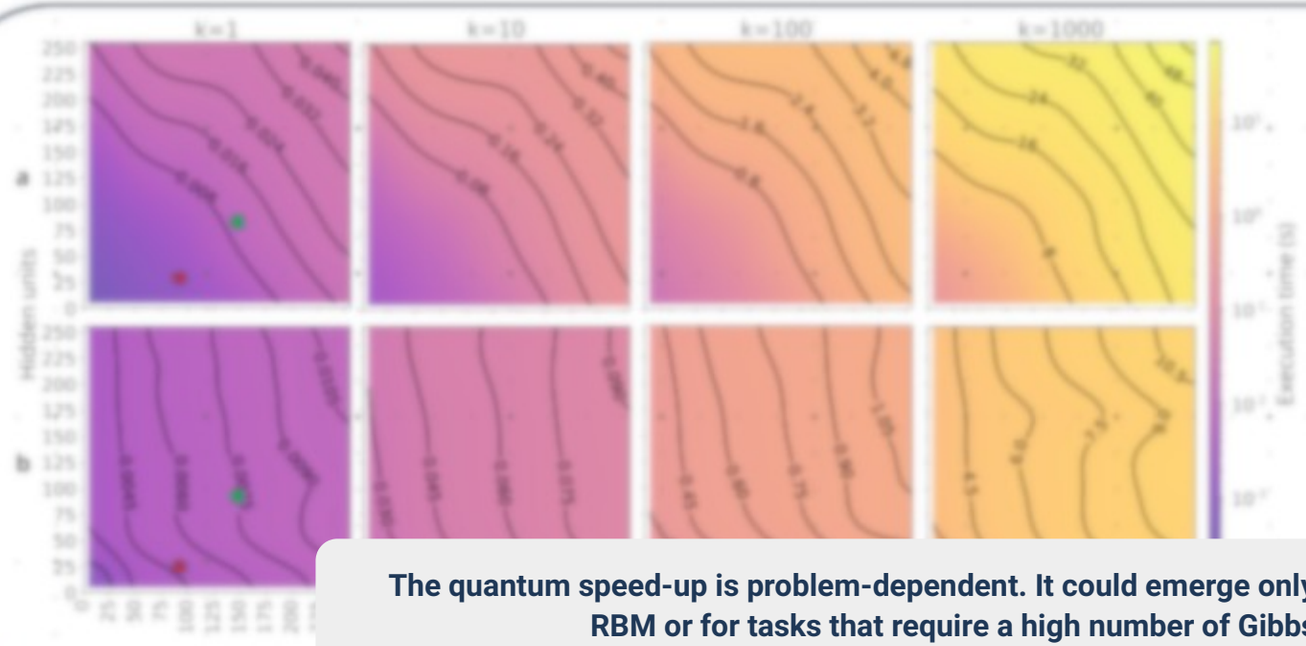


Advantage 4.1 QPU times

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

Computational Times

RBM



Single core CPU times

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

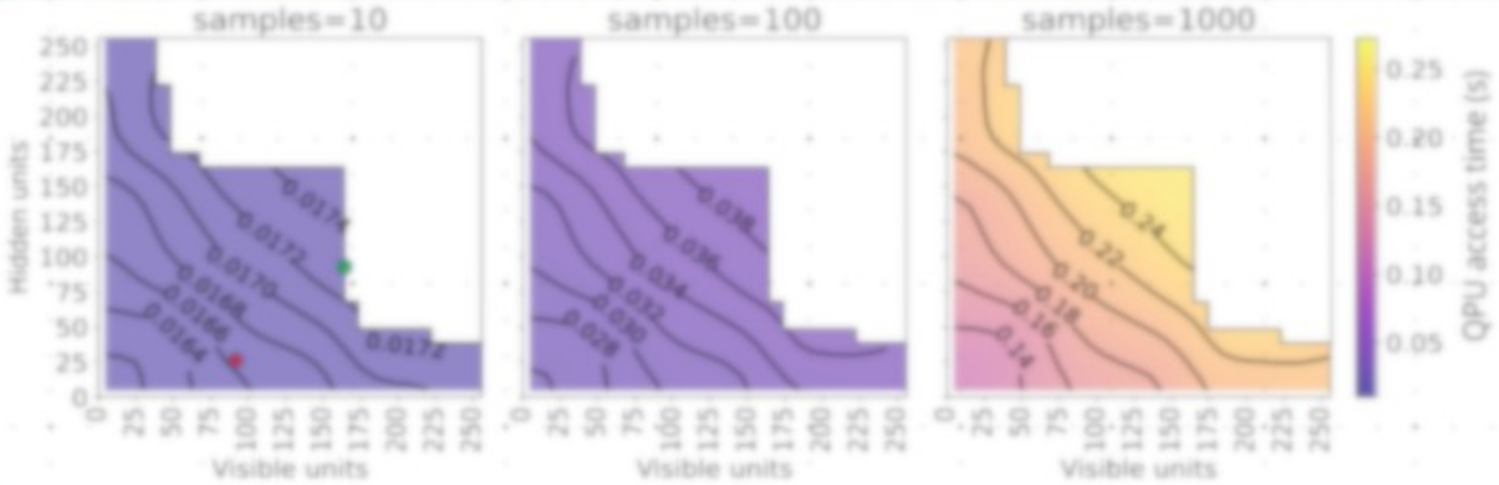
Classic times depend on the number of contrastive divergence steps (K) and number of qubits involved

128 cores CPU times

- CSE-CIC-IDS2018
- NSL-KDD

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.

QRBM



Advantage 4.1 QPU times

Quantum times depend on the number of sample extracted arid number of qubits involved

Inference Times

Dataset	k	Accuracy	F1	TP	FP	FN	TN
NSL-KDD	1	0.906 ± 0.005	0.901 ± 0.005	0.909 ± 0.009	0.091 ± 0.009	0.096 ± 0.005	0.903 ± 0.005
	10	0.935 ± 0.002	0.932 ± 0.002	0.936 ± 0.003	0.064 ± 0.003	0.065 ± 0.003	0.935 ± 0.003
	100	0.937 ± 0.002	0.934 ± 0.002	0.939 ± 0.004	0.061 ± 0.004	0.064 ± 0.001	0.935 ± 0.001
	1000	0.938 ± 0.002	0.935 ± 0.002	0.979 ± 0.002	0.020 ± 0.003	0.064 ± 0.002	0.936 ± 0.002
CSE-CIC-IDS2018	1	0.800 ± 0.004	0.805 ± 0.005	0.833 ± 0.003	0.166 ± 0.004	0.234 ± 0.006	0.766 ± 0.006
	10	0.897 ± 0.001	0.903 ± 0.001	0.907 ± 0.002	0.093 ± 0.002	0.113 ± 0.002	0.887 ± 0.002
	100	0.915 ± 0.001	0.921 ± 0.001	0.922 ± 0.003	0.078 ± 0.003	0.092 ± 0.001	0.907 ± 0.001
	1000	0.924 ± 0.001	0.929 ± 0.001	0.932 ± 0.001	0.068 ± 0.002	0.084 ± 0.001	0.916 ± 0.001

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

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

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

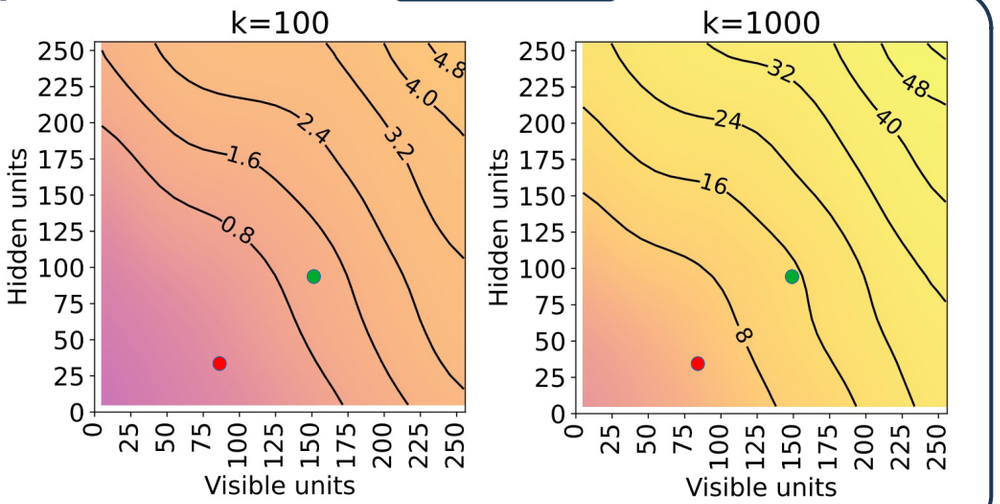
Dataset	k	Accuracy	F1	TP	FP	FN	TN
NSL-KDD	1	0.906 ± 0.005	0.901 ± 0.005	0.909 ± 0.009	0.091 ± 0.009	0.096 ± 0.005	0.903 ± 0.005
	10	0.935 ± 0.002	0.932 ± 0.002	0.936 ± 0.003	0.064 ± 0.003	0.065 ± 0.003	0.935 ± 0.003
	100	0.937 ± 0.002	0.934 ± 0.002	0.939 ± 0.004	0.061 ± 0.004	0.064 ± 0.001	0.935 ± 0.001
	1000	0.938 ± 0.002	0.935 ± 0.002	0.979 ± 0.002	0.020 ± 0.003	0.064 ± 0.002	0.936 ± 0.002
CSE-CIC-IDS2018	1	0.800 ± 0.004	0.805 ± 0.005	0.833 ± 0.003	0.166 ± 0.004	0.234 ± 0.006	0.766 ± 0.006
	10	0.897 ± 0.001	0.903 ± 0.001	0.907 ± 0.002	0.093 ± 0.002	0.113 ± 0.002	0.887 ± 0.002
	100	0.915 ± 0.001	0.921 ± 0.001	0.922 ± 0.003	0.078 ± 0.003	0.092 ± 0.001	0.907 ± 0.001
	1000	0.924 ± 0.001	0.929 ± 0.001	0.932 ± 0.001	0.068 ± 0.002	0.084 ± 0.001	0.916 ± 0.001

The number of CD steps during inference hugely affect the performance of the model

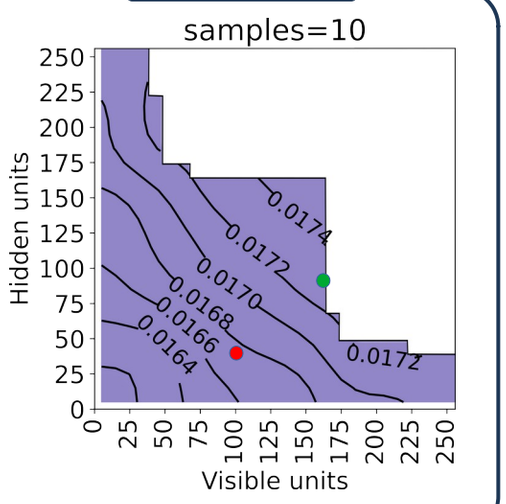
During the training it is ok having a noisy gradient ascent step

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

Classic Time



Quantum Time



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

Dataset	Processor	k	Computational Time	Speed-up
CSE-CIC-IDS2018	Single core	100	1.1 s	64x
	128 cores	100	0.71 s	41x
	QPU	10	0.017 s	1x
NSL-KDD	Single core	10	0.070 s	4x
	128 cores	10	0.035 s	2x
	QPU	10	0.016 s	1x

CSE-CIC-IDS2018

NSL-KDD

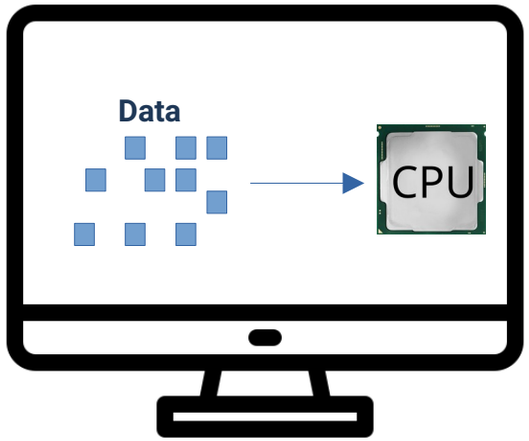
We detected a quantum speedup in the query time

However...

Cloud latency

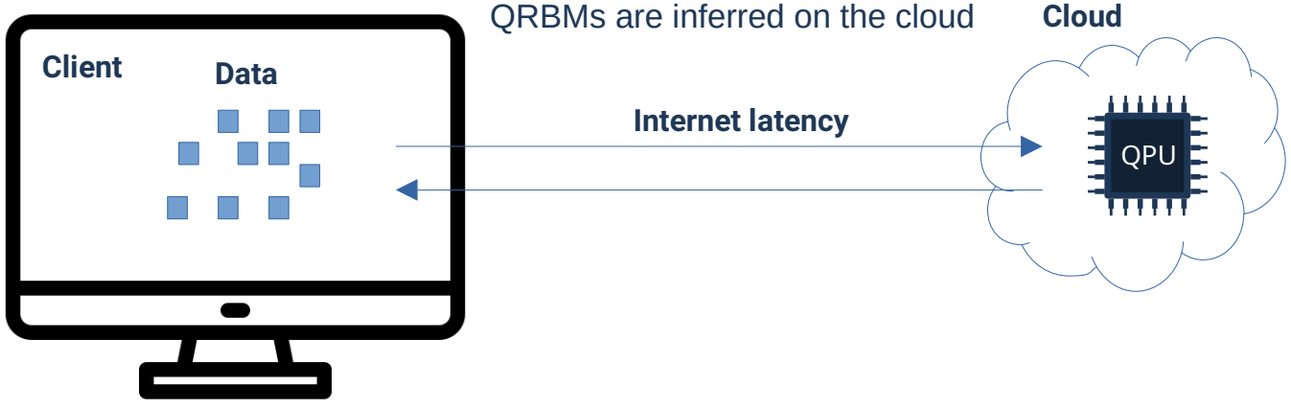
CPU

RBMs are inferred locally



QPU

QRBMs are inferred on the cloud

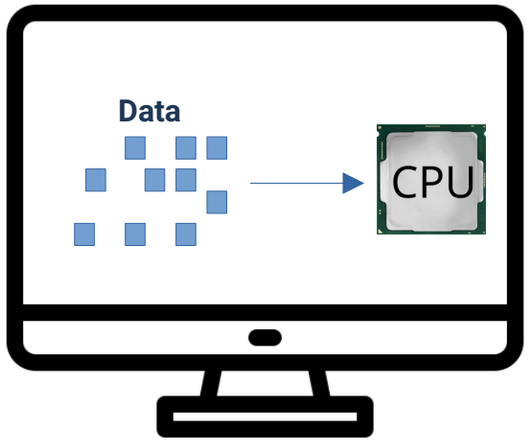


However...

Cloud latency

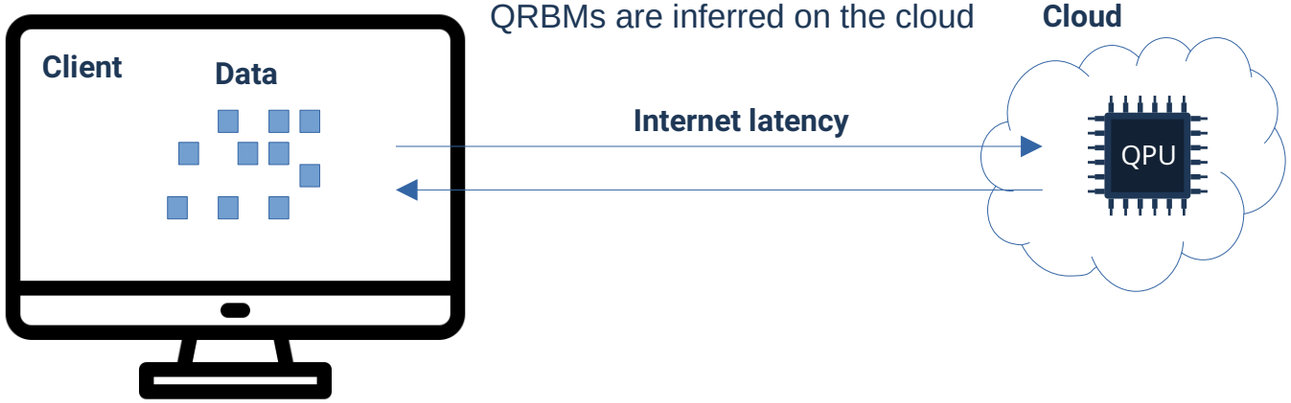
CPU

RBMs are inferred locally



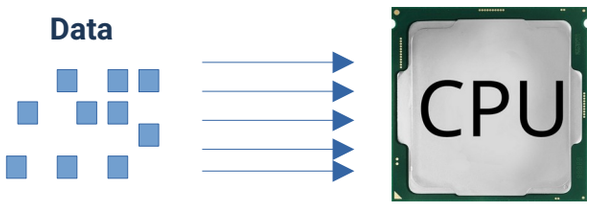
QPU

QRBMs are inferred on the cloud

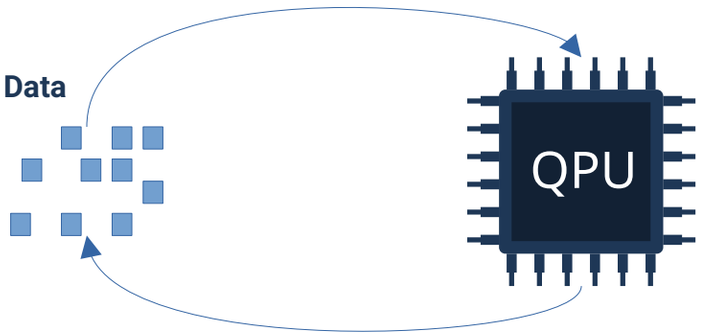


"Single core" QPU

The CPU can process data in batches



The QPU have to process on data at the time



CONCLUSION

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

QRBMs doesn’t present a computational complexity advantage on current quantum hardware (connettivity problem)

QRBMs haven't showen 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)

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 latancy 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?

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

QRBMs doesn’t present a computational complexity advantage on current quantum hardware (connettivity problem)

QRBMs haven't showen 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)

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 latancy 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