Birla Institute of Science and Technology

Project Report: Quantum Annealing On DWAVE System

Submitted To

Professor Ashwin Srinivasan

Submitted By:

Khyati Jain

In partial fulfillment of the requirements of:

CS F266: STUDY PROJECT

Quantum Annealing On DWAVE Systems

Khyati Jain

BITS Pilani Goa Campus

December, 2019

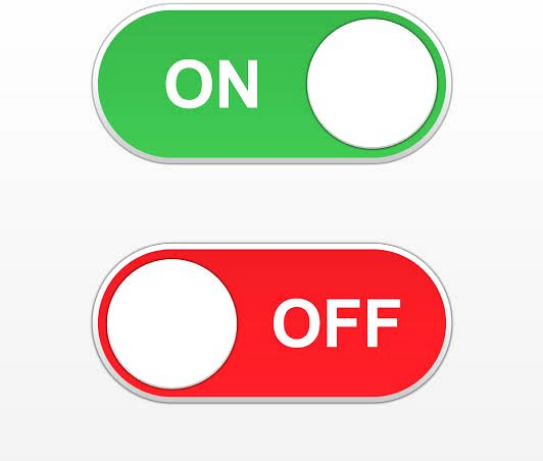
What is a Quantum Computer?

Extremely Simplified:

- It’s not a turing machine!
- Manipulates qubits instead of bits
- Harnesses quantum resource - entanglement and superposition

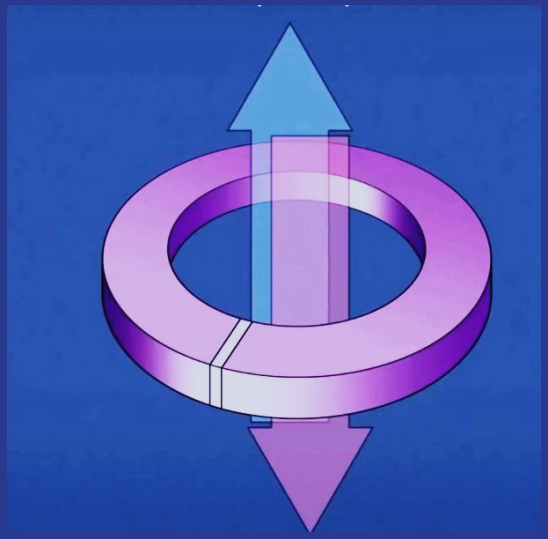
Bit

Binary Variable:
Exits in 0 or 1 state



Qubit

Superposition:
A qubit is both 0 and 1 at the same time



Candidates For Quantum Computing

- Gate-model quantum computing
  - Implemented by IBM
- Quantum Annealing
  - Implemented by DWAVE

Quantum Annealing

- Energy minimization problem implemented via cooling
- Energy diagram changes over time as the quantum annealing process runs and a bias is applied
- At the end of the quantum annealing process, each qubit collapses from a superposition state into either 0 or 1 (a classical state)

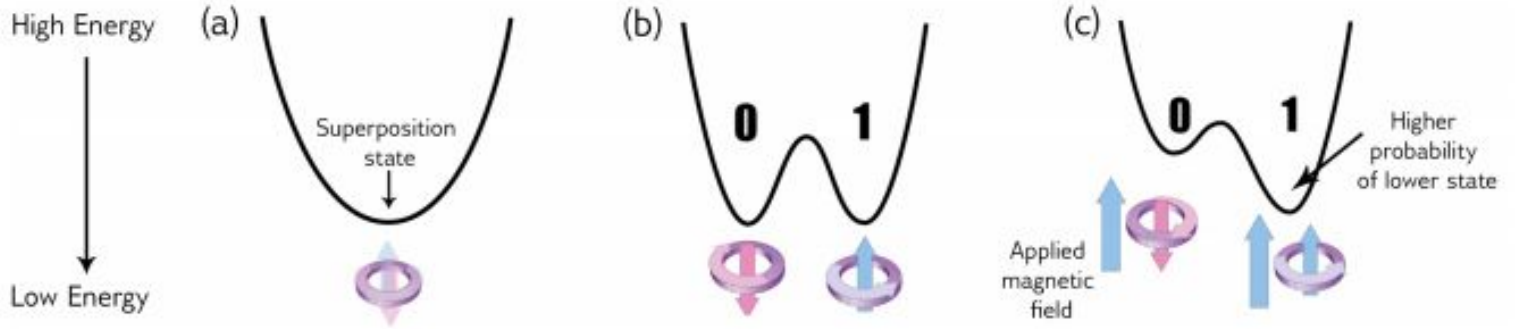


Figure: Quantum annealing process

Applicable Problems

- **Optimization Problems:**
  - require the real minimum energy by minimising the cost function
  - useful for discrete combinatorics problems
- **Probabilistic sampling problems:**
  - require good low-energy samples for characterizing the shape of the energy landscape
  - useful for machine learning problems

Problem Formulation

- **Ising Model**

$$E_{ising}(s) = \sum_{i=1}^{N} h_i s_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} J_{i,j} s_i s_j$$

where $x_i \in \{-1, 1\}$ corresponding to spin up and spin down states

- **QUBO: Q**uadradic **U**nconstrained **B**inary **O**ptimization

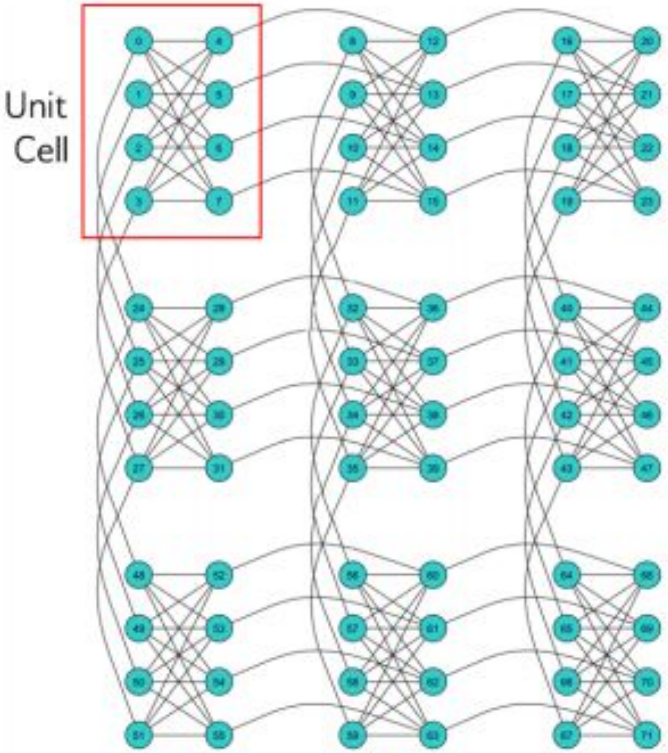
$$f(x) = \sum_{i} Q_{i,i} x_i + \sum_{i<j} Q_{i,j} x_i x_j$$

where $x_i \in \{1, 0\}$ corresponding to true and false.

Notation Conventions

Problem Expression	Terms			
	Linear coefficient	Quadratic coefficient	Variable	States
QUBO (scalar)	a_i	$b_{i,j}$	q_i	$\{0, 1\}$
QUBO (matrix)	$Q_{i,i}$	$Q_{i,j}$	x_i	$\{0, 1\}$
Ising	h_i	$J_{i,j}$	s_i	$\{-1, 1\}$
Graph	Node Weight	Edge Strength	Node	
QPU	Qubit Bias	Coupling Strength	Qubit State	{Spin Up, Spin Down}

DWAVE QPU Architecture



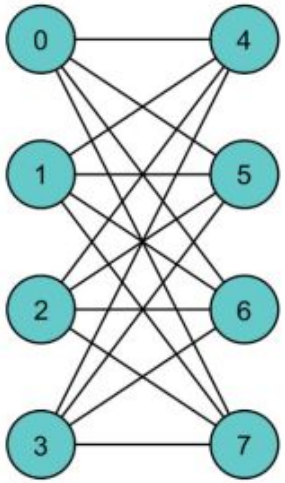
The basic unit of the DWAVE QPU Architecture is the Chimera Graph. It is a complete bipartite graph ($K_{4,4}$)

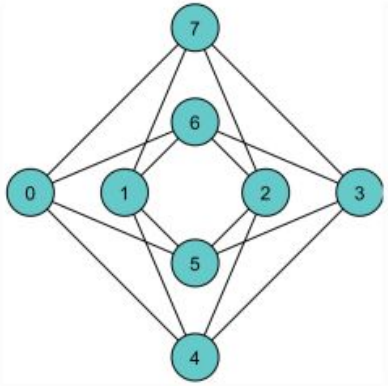
DWAVE QPU Architecture II

Chimera graphs (unit cell) have two isomorphic configurations - the column and the cross.

Each node represents a physical qubit.

DWAVE's QPU comprises of lattice of repeating unit cells, with additional connectivity between nodes of adjacent cells.





Programming on DWAVE

1. Formulate the problem in QUBO or Ising form
2. Map the states to a physical qubit
3. Set the biases and weights in the equation
4. Allow the annealing process to occur under the *influence* of these weights and biases
5. Read out the values of the qubits

Simple Optimization Problem

Task: Maximize XNOR of two qubits q_1 and q_2

- **Objective Function:** $f(s) = a_1 q_1 + a_2 q_2 + b_{1,2} q_1 q_2$ where s is a vector or the variables $q = [q_i, q_2]$, a_1 and a_2 are the qubit biases and $b_{1,2}$ is the strength of the coupler.
- We want to penalize $(0,1),(1,0)$ while strongly favouring $(1,1) and (0,0)$ equally.
- Set $a_1 = a_2 = a$ and $b_{1,2} = -2a$
- If auto-scaling is turned off, magnitude of a affects the probabilities of the outcomes.

Representing Constraints in QUBO I

Exactly-one-true constraints: to find a function E (a, b, c) that is at a minimum when this objective is true. Because the variables a binary, $a^2 = a$.

$$E(a, b, c) = (a + b + c - 1)^2 \quad (1)$$

$$E(a, b, c) = 2ab + 2ac + 2bc - a - b - c + 1 \quad (2)$$

Representing Constraints in QUBO II

- Truth Table for Exactly-one-true-constraint:

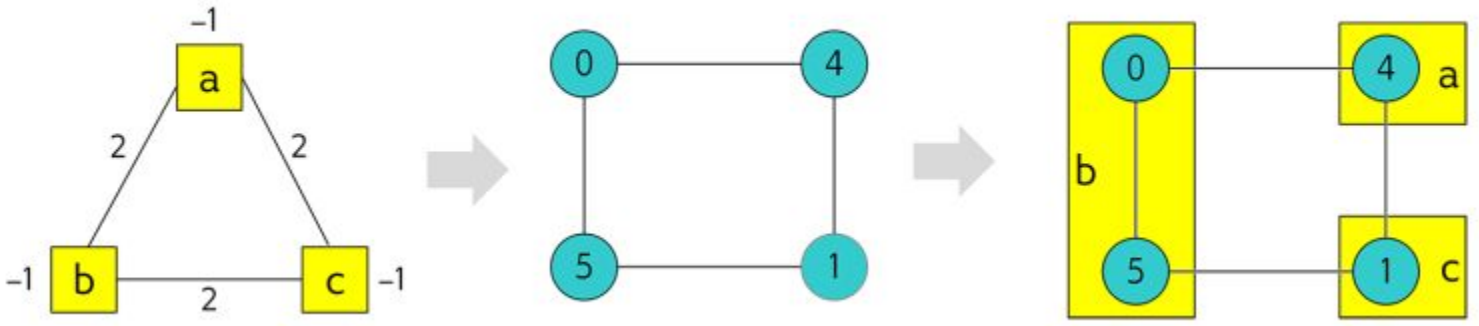
a	b	c	Exactly 1	Energy
0	0	0	FALSE	1
1	0	0	TRUE	0
0	1	0	TRUE	0
1	1	0	FALSE	1
0	0	1	TRUE	0
1	0	1	FALSE	1
0	1	1	FALSE	1
1	1	1	FALSE	4

- When expressed as QUBO we obtain:

$$E(x_0, x_1, x_2) = 2x_0x_1 + 2x_0x_2 + 2x_1x_2 - x_0 - x_1 - x_2 + 1.$$

Minor Embedding the Problem

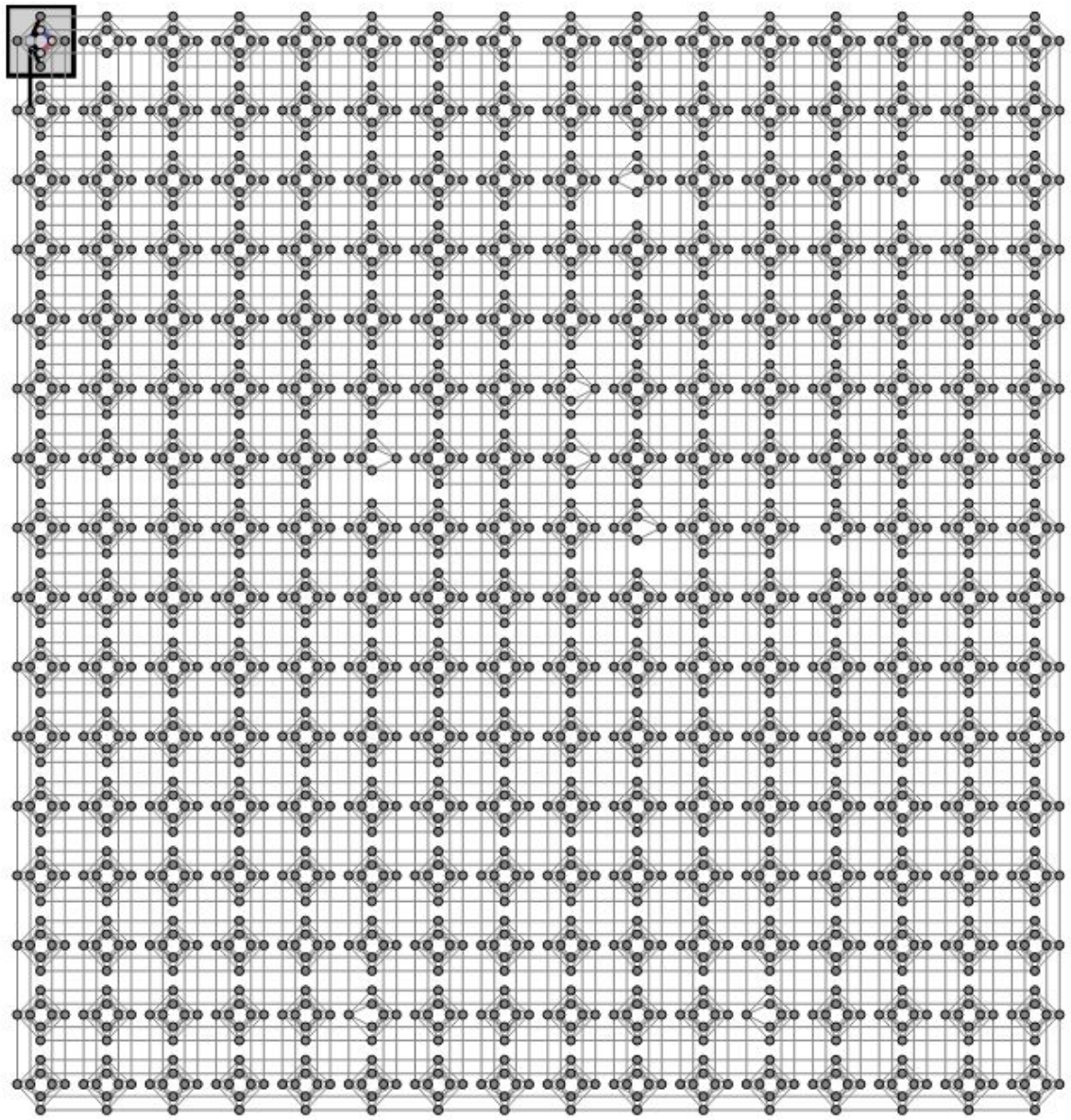
- Mapping from variables to qubits is known as minor embedding
- For this problem: Need to fit a 3 -qubit loop into a 4-sided structure

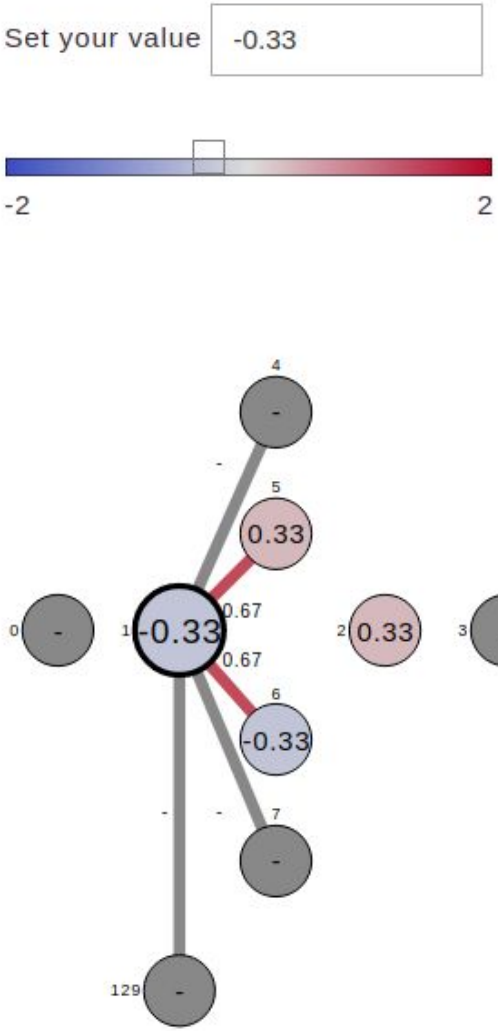


(Image Retrieved From DWAVE Documentation : Minor Embedding)

- Qubit 0 is chained to qubit 5 to represent variable b.

Running on DWAVE Visual Solver I





(Screenshot of DWAVE's visual solver)

Running on DWAVE Visual Solver II

Solution	Energy	Occurrences
[0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...] Graph Show	-0.34	33
[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...] Graph Show	-0.33	42
[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...] Graph Show	-0.33	25

The solution shows the three possible answers (1,0,0), (0,1,0), and (0,0,1)

Map Coloring Problem

Map Coloring Problem

Objective: Given any separation of a plane into contiguous regions, objective is to map each region to a color $c, c \in \mathbb{C}$ such that no two regions which share a boundary have the same color and each region is mapped to only one color.

- Each region is mapped to only one color, of C possible colors.
- No two regions which share a boundary have the same color

Solving the problem means finding a permissible color for each of the regions.

Expressing the constraints in binary variables

- Use unary encoding. (Number of bits = number of colors)
- For C=4:

Color	Naturals	Unary Encoding
Blue	1	$q_B, q_G, q_R, q_Y = 1, 0, 0, 0$
Green	2	$q_B, q_G, q_R, q_Y = 0, 1, 0, 0$
Red	3	$q_B, q_G, q_R, q_Y = 0, 0, 1, 0$
Yellow	4	$q_B, q_G, q_R, q_Y = 0, 0, 0, 1$

Expressing the constraints in binary variables II

- For a two-color problem: the constraint that a region be assigned a single color only.

$$\mathrm{E}(a_i, b_{i,j}; q_i) = a_B q_B + a_G q_G + b_{B,G} q_B q_G.$$

- Solution: set $a_B = a_G = -1$ and $b_{B,G} = 2$. This sets the minimum energy for this objective to -1 for both valid states and 0 for invalid states

q_B	q_G	Constraint	$\mathrm{E}(a_i, b_{i,j}; q_i)$
0	0	Violates	0
0	1	Meets	a_G
1	0	Meets	a_B
1	1	Violates	$a_B + a_G + b_{B,G}$

- Similarly for coupling two regions:

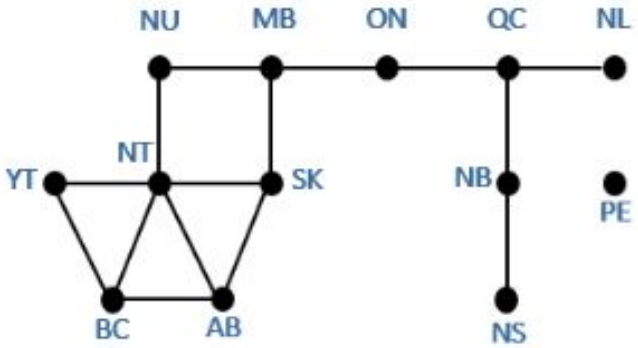
$$\mathrm{E}(a_i, b_{i,j}; q_i) = a_R^{BC} q_R^{BC} + a_R^{AB} q_R^{AB} + b_R^{BC,AB} q_R^{BC} q_R^{AB}$$

Where BC and AB denote two regions.

Problem Formulation

For example, take the Map of Canada:

Embedding onto a graph



AB Alberta
BC British Columbia
MB Manitoba
NB New Brunswick
NL Newfoundland and Labrador
NS Nova Scotia
NT Northwest Territories
NU Nunavut
ON Ontario
PE Prince Edward Island
QC Quebec
SK Saskatchewan
YT Yukon

Image and Example reproduced from: E. D. Dahl, Programming with D-Wave: Map Coloring Problem, November 2013

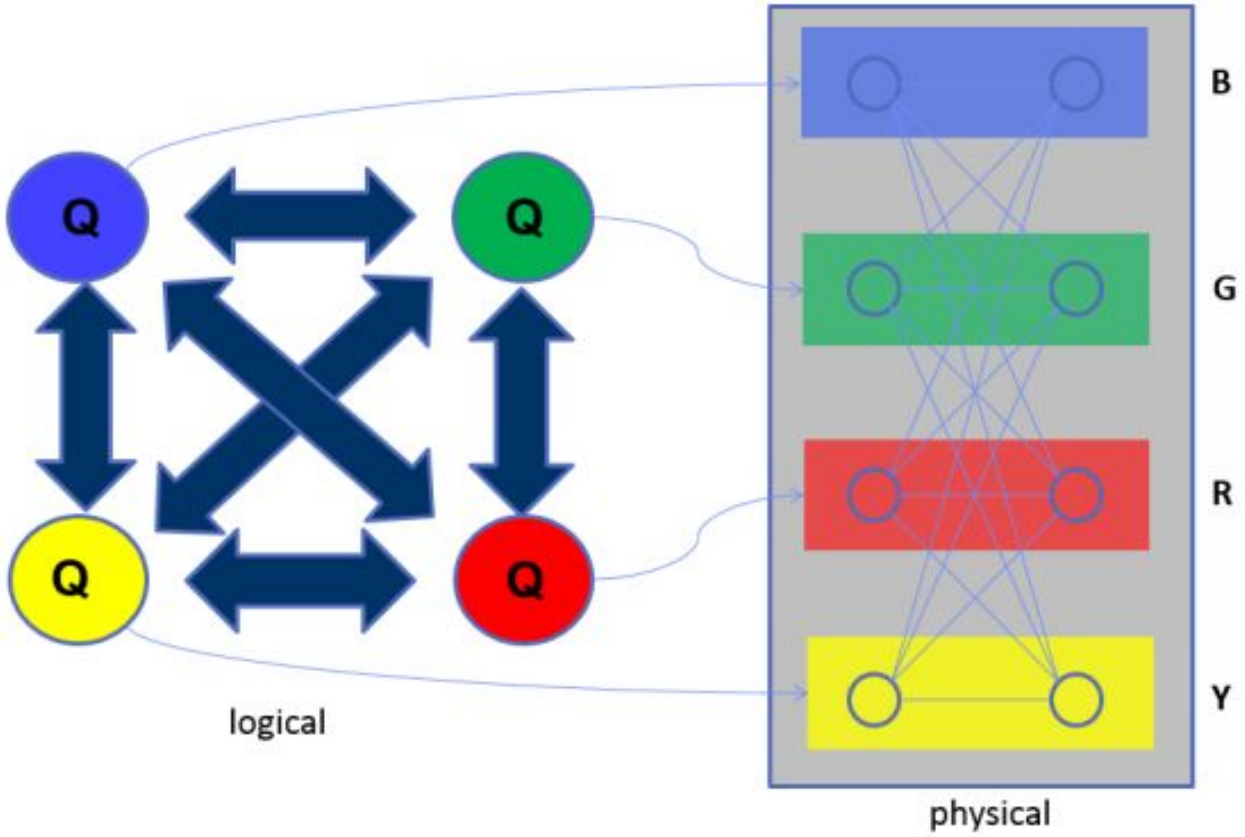
Minor Embedding a region I

Problem: The constraints of a four-color map problem require full connectivity between the four qubits encoding the color for each region. Not available in Chimera Graph.

Solution: We need to embed the logical qubits to physical qubits through *chaining.*

Minor Embedding a region II

Mapping logical qubits to physical qubits:



Expressing constraints in Binary Variables III

- A chain is formulated as a constraint: All the physical qubits should have identical spin to represent one logical qubit.

$$\mathrm{E}(a_i, b_{i,j}; q_{Bi}) = a_1 q_{B1} + a_2 q_{B2} + b_{1,2} q_{B1} q_{B2}$$

  where q_{Bi} are the two physical qubits of the logical qubit for blue.

- Solution : set a1 = a2 = 1 and b1,2 = − 2. This sets the minimum energy for this objective to 0 for both valid states and 1 for the states that violate the constraint.

Coupling two regions

Problem: Not all adjacent regions can be directly embedded onto the QPU

Solution: Clone regions as required

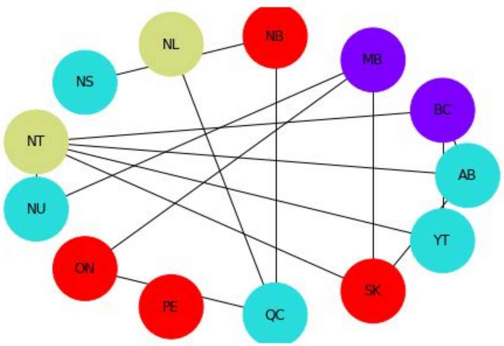
Example: Map of Canada :

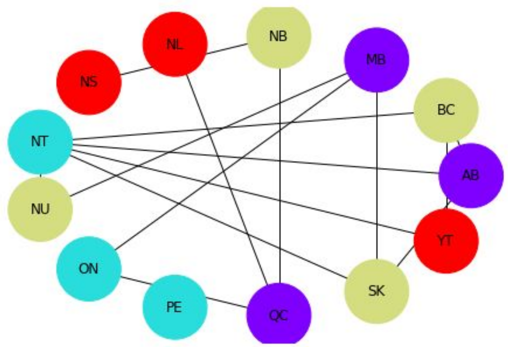
	0	1	2	3	4
0	NL	ON	MB	SK	AB
1	PE	QC	NU	NT	AB
2		NB	NS	NT	BC
3				YT	BC

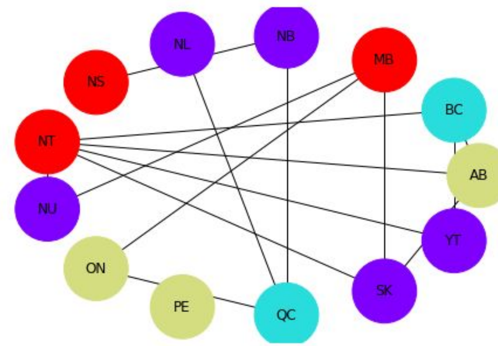
Results - Canada’s Map

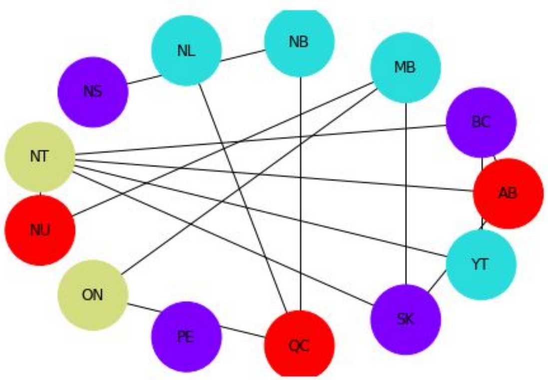
For the Canada Map problem : 37 failures in 50 runs

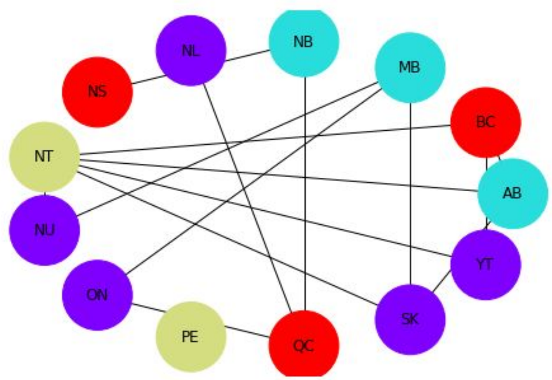
Following graphs represent the connectivity and colors obtained in a successful run:





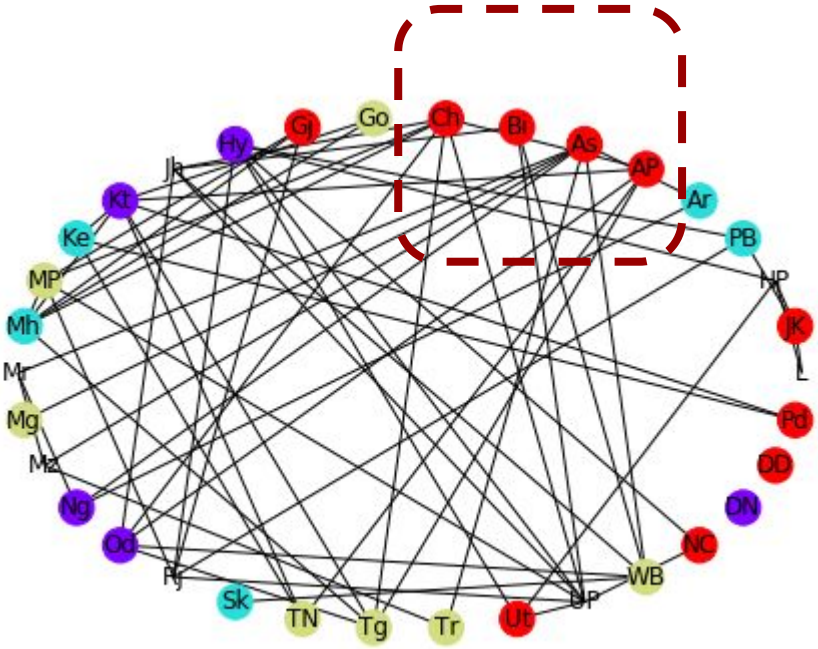




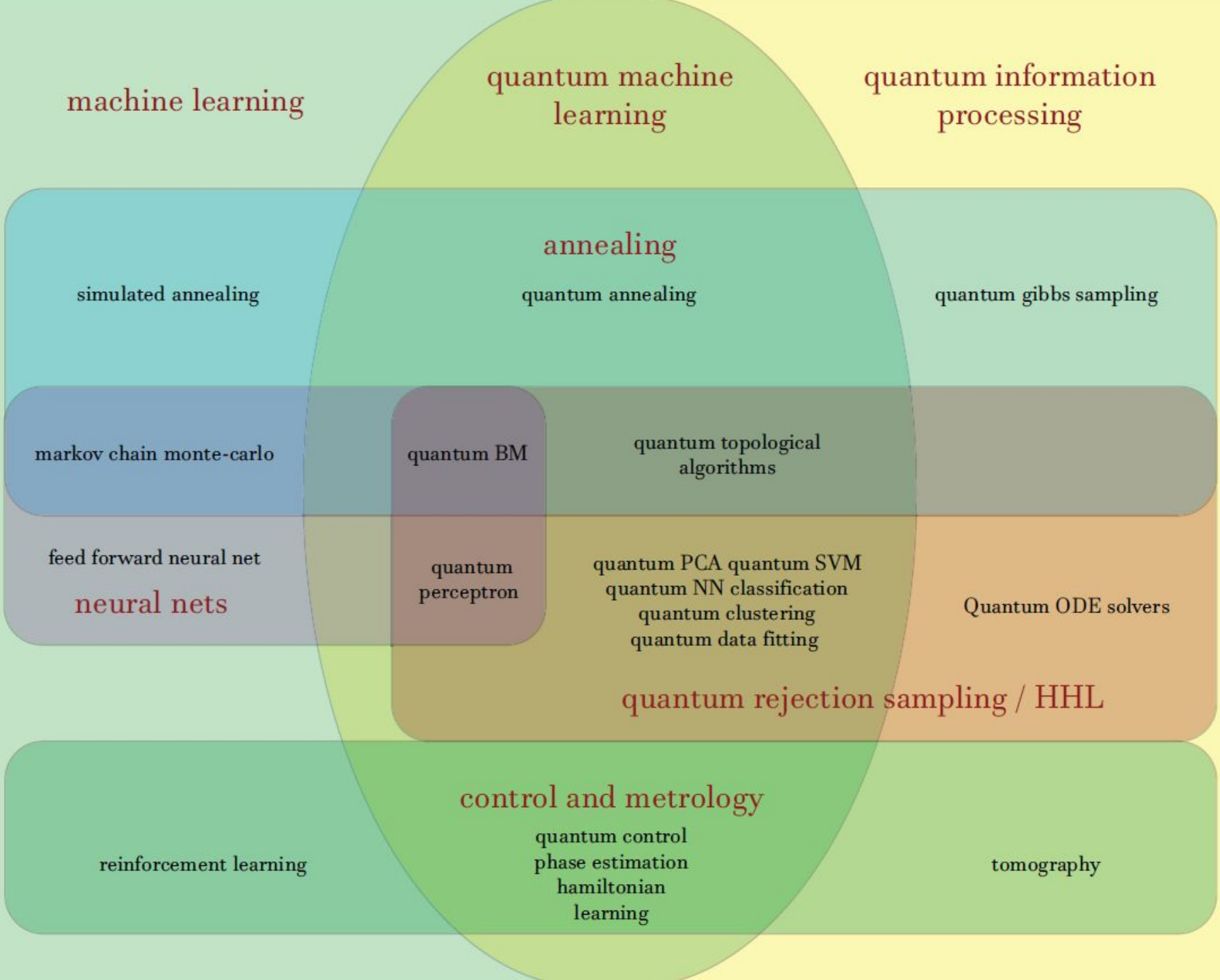


Results- India's Map

- More complex : 35 Nodes, 62 Edges
- Fails to anneal to the correct result in 200 attempts



Quantum Machine Learning on DWAVE System

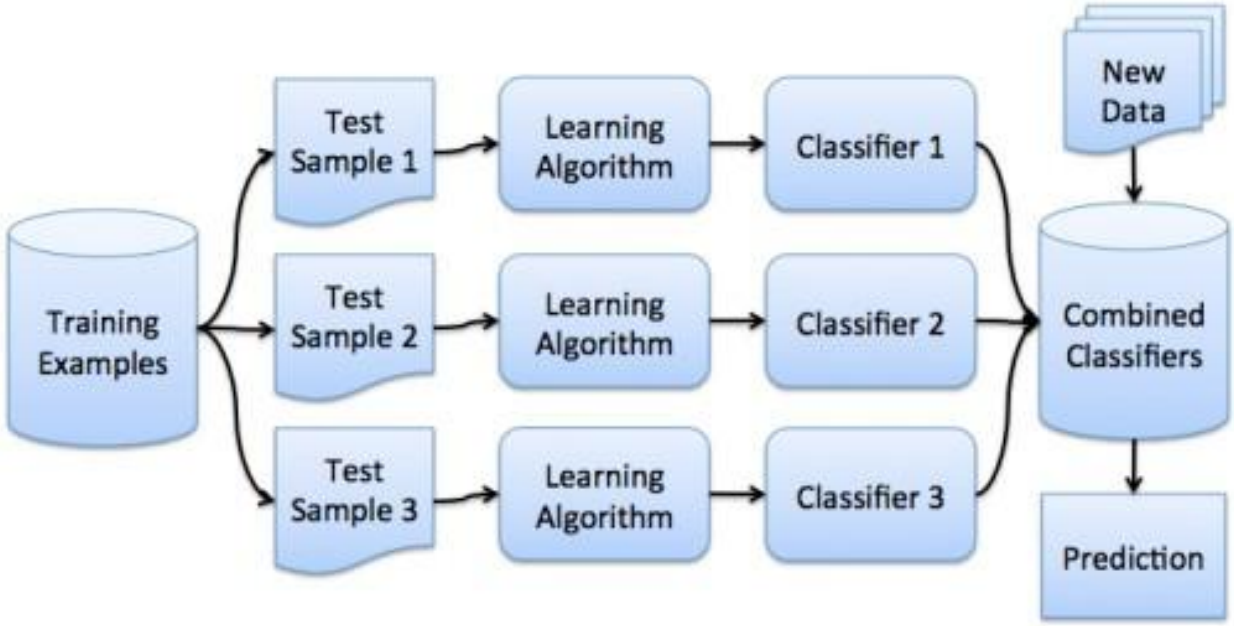

machine learning
quantum machine learning
quantum information processing
annealing
simulated annealing
quantum annealing
quantum gibbs sampling
markov chain monte-carlo
quantum BM
quantum topological algorithms
feed forward neural net
neural nets
quantum perceptron
quantum PCA quantum SVM
quantum NN classification
quantum clustering
quantum data fitting
Quantum ODE solvers
quantum rejection sampling / HHL
control and metrology
reinforcement learning
quantum control
phase estimation
hamiltonian learning
tomography

QBoost Algorithm

- Is an ensemble method which binary classification amenable to quantum computing
- Problem is formulated as a thresholded linear superposition of a set of weak classifiers
- The D-Wave is used to optimize the weights in a learning process that minimizes the training error and number of weak classifiers.

Preliminaries -Ensemble Methods

Ensemble methods build a strong classifier by combining weak classifiers



Boosting

- Weak Learner ⟶ Strong Learner

- Identify weak rules :

  Apply base learning (ML) algorithms with a different distribution

- Choose different distribution for each round :

  The base learner takes all the distributions and assigns initial equal weight to each observation. Iteratively: Pay higher attention to observations having prediction error. Then, apply the next base learning algorithm to improve accuracy.

AdaBoost

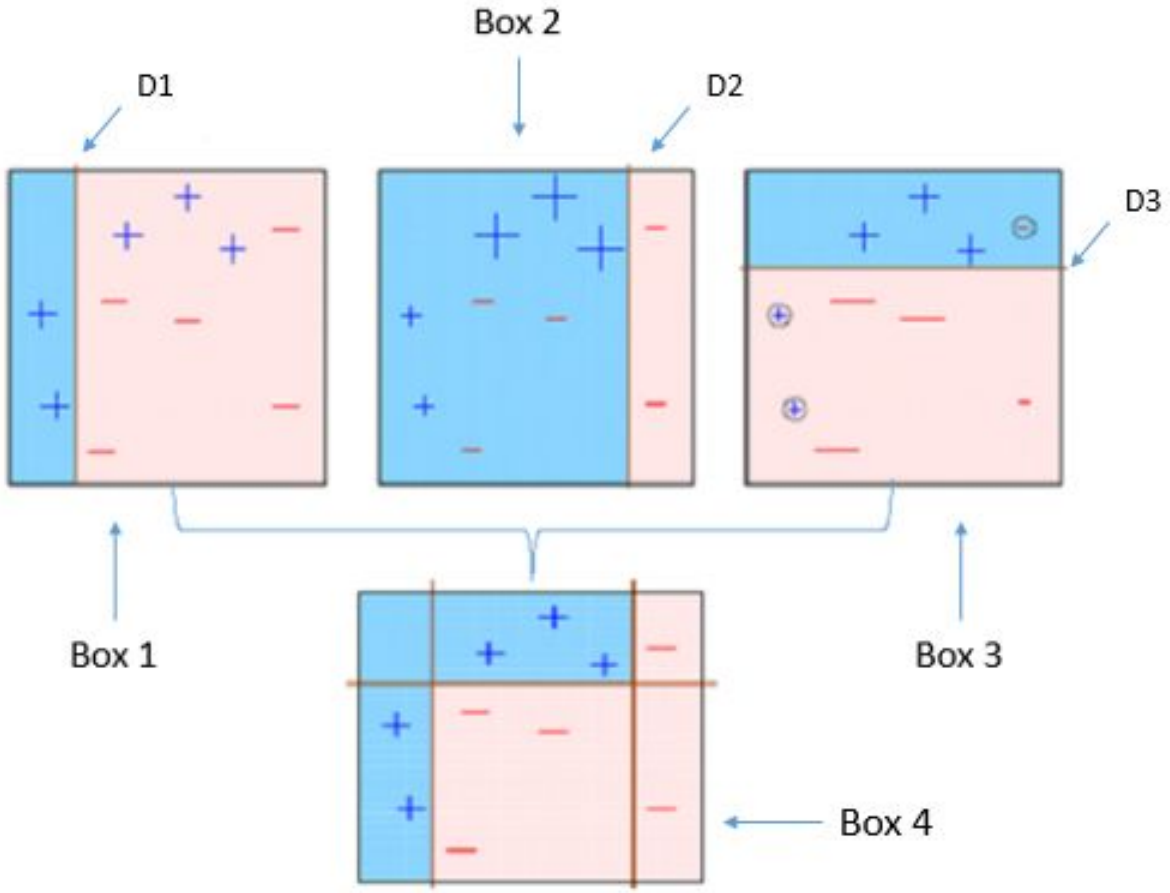


Image obtained from this introductory article on AdaBoost:
https://towardsdatascience.com/understanding-adaboost-2f94f22d5bfe

Binary Classification using Boosting

Classifier:

$$y = H(x) = \text{sign}\left(\sum_{i=1}^{N} w_i h_i(x)\right)$$

where $x \in \mathbf{R}^M$ are the input patterns to be classified

$y \in \{-1, 1\}$ is the output of the classifier

$h_i : x \longrightarrow \{-1, 1\}$ are weak classifiers or features detectors

$w_i \in [0, 1]$ are a set of weights to be optimized.

$H(x)$ is known as a strong classifier.

Binary Classification using Boosting II

Loss function:

- A natural choice is 0-1 error, which counts the number of misclassifications over the training set.

$$L(w) = \sum_{s=1}^{S} \mathbf{H}\left(-y_s \sum_{i=1}^{N} w_i h_i(x_s)\right)$$

- **H** is the Heaviside step function.
- Regularization Term:

$$R(w) = \lambda \parallel w \parallel_0 = \lambda \sum_{i=1}^{N} {w_i}^0$$

So, the problem reduces to the following minimization problem:

$$\begin{aligned} w^{opt} &= \arg\min_w \left(L(w) + R(w)\right) \\ &= \arg\min_w \left(\sum_{s=1}^{S} \mathbf{H}(-y_s \sum_{i=1}^{N} w_i h_i(x_s)) + \lambda \sum_{i=1}^{N} {w_i}^0\right) \end{aligned}$$

QBoost I

Formulating the Problem in QUBO:

- Need to transition from continuous weights $w_i \in [0, 1]$ to binary variables ⇒ a binary expansion of the weights.

  It turns out we need only a few bits! (often a single bit suffices)

- Bit constraining can be regarded as an intrinsic regularization
- Using quadratic Loss:

$$w^{opt} = \arg\min_w \left(\sum_{s=1}^{S} | \sum_{i=1}^{N} w_i h_i(x_s) - y_s|^2 + \lambda \parallel w \parallel_0 \right)$$

$$= \arg\min_w \left(\sum_{s=1}^{S} \left(\left(\sum_{i=1}^{N} w_i h_i(x_s) \right)^2 - 2 \sum_{i=1}^{N} w_i h_i(x_s) y_s + y_s^2 \right) + \lambda \sum_{i=1}^{N} {w_i}^0 \right)$$

$$= \arg\min_w \left(\sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j \underbrace{\left(\sum_{s=1}^{S} h_i(x_s) h_j(x_s) \right)}_{Corr(h_i, h_j)} + \sum_{i=1}^{N} w_i \left(\lambda - 2 \underbrace{\sum_{s=1}^{S} h_i(x_s) y_s}_{Corr(h_i, y)} \right) \right)$$

QBoost II

Formulating the Problem in QUBO:

- Using quadratic Loss and assuming 1 bit for w:

$$w^{opt} = \arg\min_{w} \left(\sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \underbrace{\left(\sum_{s=1}^{S} h_i(x_s) h_j(x_s) \right)}_{Corr(h_i,h_j)} + \sum_{i=1}^{N} w_i \left(\lambda - 2 \underbrace{\sum_{s=1}^{S} h_i(x_s) y_s}_{Corr(h_i,y)} \right) \right)$$

Coupling term

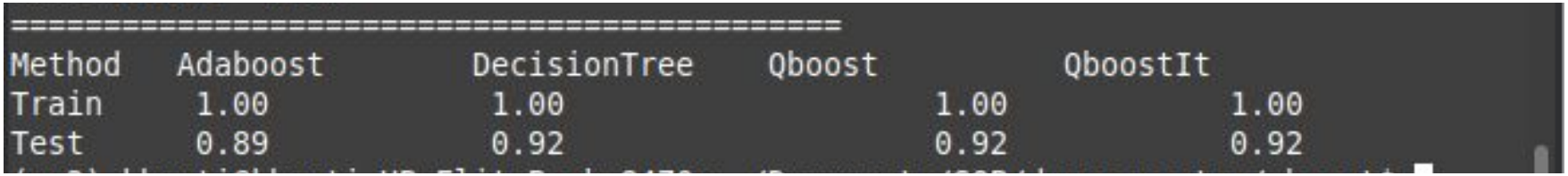
Bias

- Bias Term: If the output of weak classifiers h_i is well correlated with the labels y the bias term is lowered, increasing the probability that $w_i = 1$.
- Coupling term: Strongly correlated weak classifiers cause the coupling energy to go up, increasing the probability for one of the correlated classifiers to be switched off (either w_i or w_j becomes 0)

QBoost - Results

Sample run: Classifies tumors in scikit-learn's Wisconsin breast cancer dataset as either malignant or benign (binary classification). Train on a random ⅔ and test on the remaining ⅓. Test error of 92%. Train#: 379, Test#: 190

- Comparison with other algorithms:



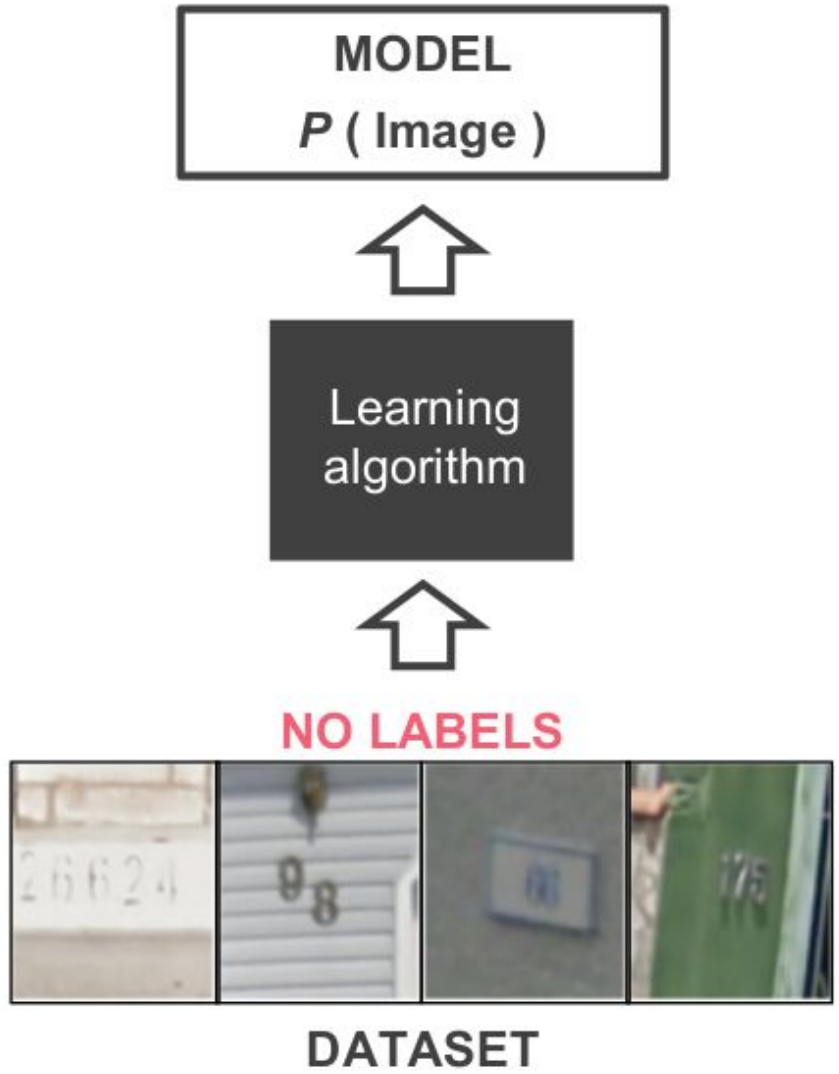
Method	Adaboost	DecisionTree	Qboost	QboostIt
Train	1.00	1.00	1.00	1.00
Test	0.89	0.92	0.92	0.92

- Competes successfully with greedy methods such as the state-of-the art method AdaBoost.

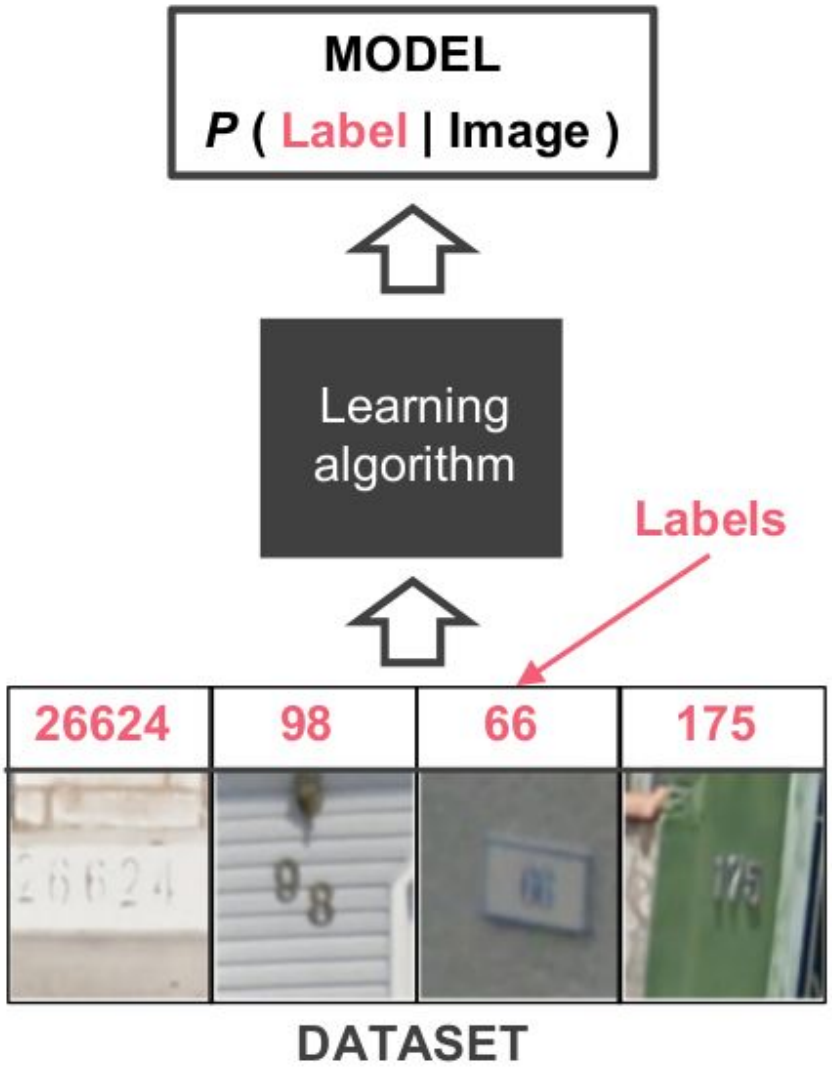
Implementing a Restricted Boltzmann Machine on DWAVE

Unsupervised Learning | Supervised Learning

Learn the "best" model distribution that can generate the same kind of data



Learn the "best" model that can perform a specific task



Probabilistic Modelling

Aim:

- Learn from noisy and unlabeled data
- Define confidence levels in predictions
- Allow decision making in the absence of complete information

How?

- *Probability distributions* represent the unobserved quantities.
- Data distributions approximated based on finite set of*samples*.
- *Learning* : *Prior* $\rightarrow$ *Posterior*

Boltzmann Distribution

- Energy-based probability distribution that defines probability p, for each of the discrete states in a binary vector.
- Let **x** represent a set of N binary random variables such that

$$x^T = [x_1, x_2, \ldots, x_N]$$

where $x_n \in 0, 1$ is the state of the n^{th} binary random variable in **x** .

- The Boltzmann distribution defines a probability distribution

$$p(x) = \frac{1}{Z} \exp(-E(x;\theta)) \tag{3}$$

where $E(x;\theta)$ is an energy function parameterized by θ and

$$Z = \sum_x \exp(-E(x;\theta)) \tag{4}$$

Z is the normalising factor.

Boltzmann Distribution II

- $E(x;\theta)$ is represented via a quadratic form $x^T Qx$ in which the matrix Q_θ is defined by the biases (q_i) and correlation weights ($q_{i,j}$).

$$E(x) = x^T Qx = \sum_{i<j} q_{i,j} x_i x_j + \sum_i q_{i,i} x_i. \tag{5}$$

- If $q_{i,j}$ is small (a negative value with a large magnitude), then x_i and x_j are more likely to be 1 at the same time. The diagonal entries of Q bias the probability of individual binary variables in x. If q_i is large, then x_i is more likely to be zero.

Boltzmann Machine I

- A Boltzmann Machine is a network of symmetrically connected, neuron-like binary units that make stochastic decisions about whether to be on or off.
- Have a simple learning algorithm that allows them to discover interesting features in datasets composed of binary vectors.

Boltzmann Machine II

Example: Constructing simple *Boltzman Machine* using two binary random variables, $x^T = [x_1, x_2]$ Assume

$$Q = \begin{bmatrix} -1 & 2 \\ 0 & -1. \end{bmatrix}$$

Then,

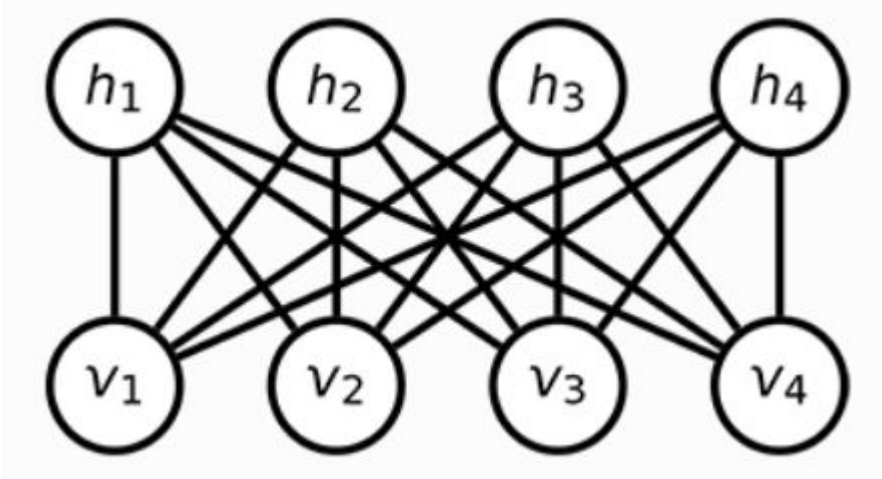
$$E(x) = x^T Q x = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} -1 & 2 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = -x_1 - x_2 + 2x_1x_2.$$

This gives the probability of each state as the following

x_1x_2	$E_Q(x)$	p(x)
0 0	0	0.13
0 1	-1	0.37
1 0	-1	0.37
1 1	0	0.13

Restricted Boltzmann Machine I

- Defines a probability distribution over a set of binary variables that are divided into visible (input) v, and hidden h, variables.
- Constraint: All visible nodes are connected to all hidden nodes. No two visible or two hidden nodes are connected (Bipartite Graph)



Restricted Boltzmann Machine II

- Learning Process: visible variable is responsible for a feature from an item in the dataset to be learned.
- Energy Function $E(x) = E(v, h)$. And $p(x; \theta) = p(v, h; \theta)$
- We are interested in: $p(v; \theta) = \sum_h p(v, h; \theta)$
- Training: Minimizing the Negative Log Likelihood of the given D training (visible) examples using gradient descent.
- The likelihood is $L(\theta) = \prod_{d=1}^{D} p(v^{(d)}; \theta)$

$$LL(\theta) = log(L(\theta)) = \sum_{d=1}^{D} \log p(v^{(d)}; \theta)$$

- Gradient descent method is used to minimize the $NLL(\theta)$

Restricted Boltzmann Machine III

- To calculate the gradient at a particular θ, we must evaluate some expected values: $E_{p(x;\theta)} f(x)$ for a set of functions $f(x)$ known as the sufficient statistics.
- BUT: The expected values cannot be determined exactly, because we cannot sum over all 2^N configurations
- We approximate by only summing over the most probable configurations, which we obtain by **sampling from the distribution** given by the current θ.

Restricted Boltzmann Machine - Hamiltonian

- N-variable system has 2^N classical states, and can be represented by the classical Ising Hamiltonian operator (matrix)

$$H = \sum_a b_a \sigma_a^z + \sum_{a,b} w_{ab} \sigma_a^z \sigma_b^z$$

where $\sigma^z = \begin{bmatrix} -1 & 0 \\ 0 & +1 \end{bmatrix}$

- This Hamiltonian is a diagonal matrix with dimensions 2^N by 2^N. The classical states are eigenvectors, and the corresponding energies are eigenvalues of the Hamiltonian.
- Boltzmann distribution: a probability distribution on the eigenvectors of a Hamiltonian such that the probability of choosing an eigenvector is related to its eigenvalue.

Quantum Boltzmann Machine

- Hamiltonian has off-diagonal terms. Eigenvector is now a superposition of classical states.
- Let s represents classical states and Ψ represents quantum states, then

$$p(s) = \sum_i p(\psi_i) |\langle s|\psi_i\rangle|^2.$$

- Hamiltonian including the *Transverse Field:*

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

Transverse Field
(quantum fluctuations)

Learning the parameters

It turns out $\frac{\partial \log p(\mathbf{v})}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}$

Hence using stochastic gradient ascent: $\Delta w_{ij} = \epsilon(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model})$

Unbiased sample of $\langle v_i h_j \rangle_{data}$:Because there are no direct connections between hidden units in an RBM, given a randomly selected training image v, the binary state h_j , of each hidden unit j, is set to 1 with probability

$$p(h_j = 1 \mid \mathbf{v}) = \sigma(b_j + \sum_i v_i w_{ij})$$

Similarly,

$$p(v_i = 1 \mid \mathbf{h}) = \sigma(a_i + \sum_j h_j w_{ij})$$

Where,

$$\sigma(x) = 1/(1 + \exp(-x))$$

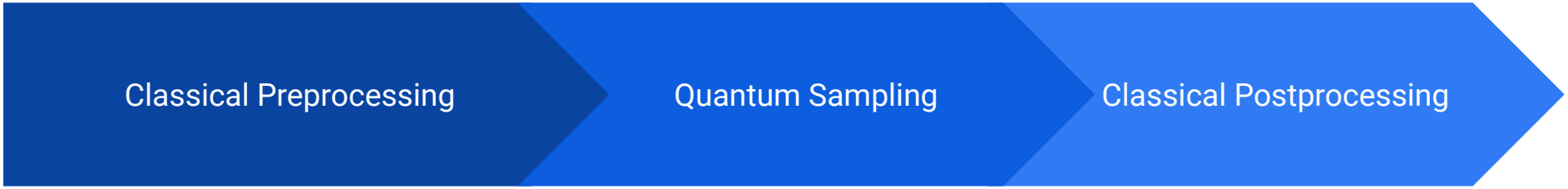
Learning weights - Classical Algorithm

Given D training examples, where each training example is a binary vector with n elements, say corresponding to a user's movie preferences. Then for each epoch, do the following:

- Set the states of the visible units to a training example.
- Update the states of the hidden units using the logistic activation rule: for the jth hidden unit,activation energy $a_j = \sum_i w_{ij} x_i$, and set x_j to 1 with probability $\sigma(a_j)$ else set to 0. For each edge e_{ij}, $Positive(e_{ij}) = x_i * x_j$.
- Reconstruct the visible units: for each visible unit, compute its activation energy a_i, and update its state. Then update the hidden units again, and compute $Negative(e_{ij}) = x_i * x_j$ for each edge.
- Update the weight of each edge e_{ij} by setting $w_{ij} = w_{ij} + L * (Positive(e_{ij}) - Negative(e_{ij}))$, where L is a learning rate.
- Repeat over all training examples.

Using DWAVE Sampling services

- Getting an unbiased sample of $\langle v_i h_j \rangle_{model}$ is difficult. (computational Bottleneck)
- We turn to Quantum *Assisted* Machine Learning.
- DWAVE samples from energy-based distributions by seeking low energy states
- These more probable values are used to calculate expectation values.



Challenges of the hybrid approach:

- Need to solve classical-quantum model mismatch.

$$\Delta w_{ij} = \epsilon(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model})$$

classical

quantum

- Robustness to noise, Fully visible models intrinsic control errors, and to deviations from sampling model (e.g., Boltzmann)
- Limited connectivity – parameter setting

Using DWAVE Sampling services

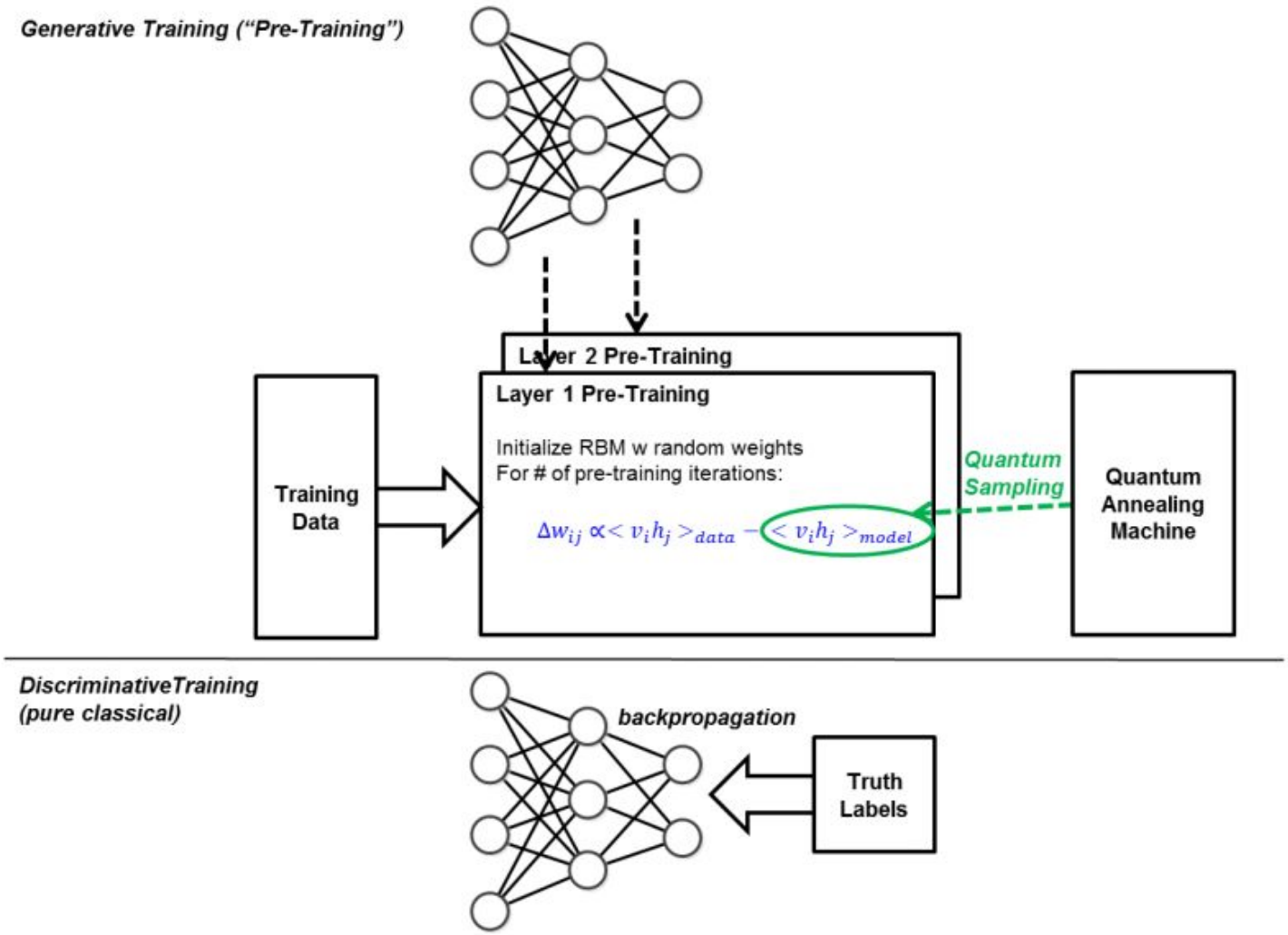
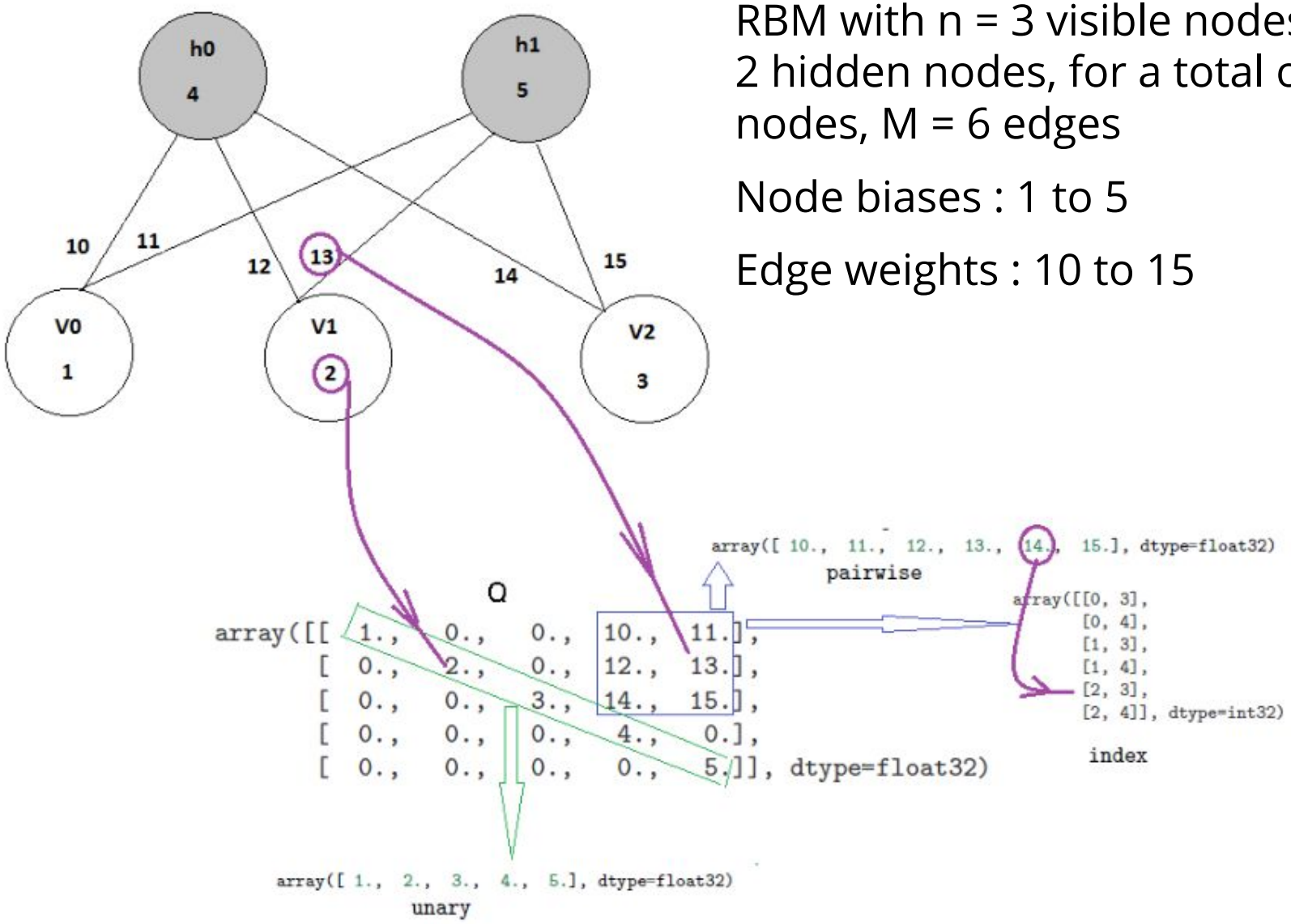


Figure 2 Overall training approach including generative and discriminative training

Figure retrieved from https://arxiv.org/pdf/1510.06356.pdf

Example of QUBO Problem Formulation



RBM with n = 3 visible nodes, m = 2 hidden nodes, for a total of N = 5 nodes, M = 6 edges

Node biases : 1 to 5

Edge weights : 10 to 15

Embedding the problem

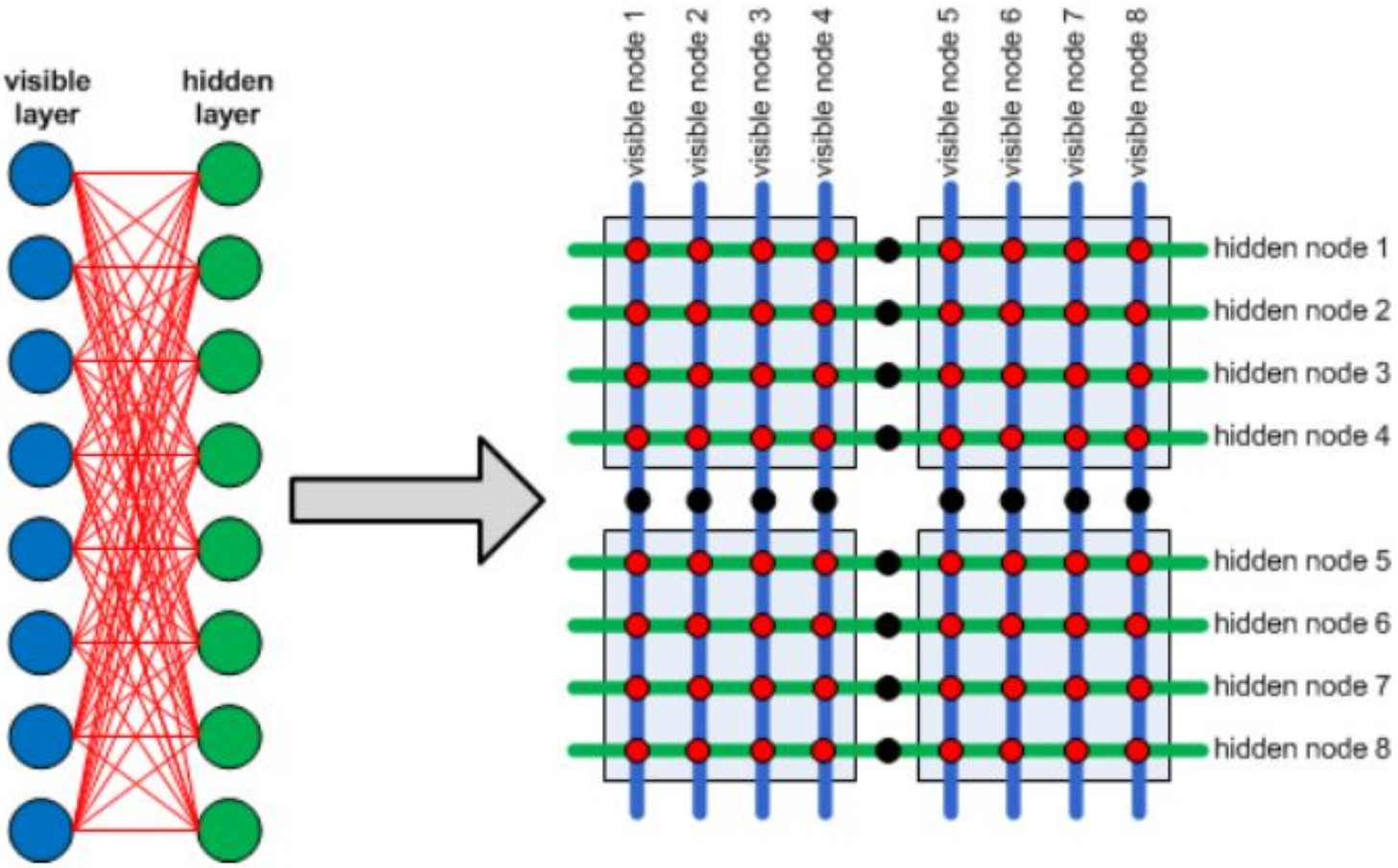
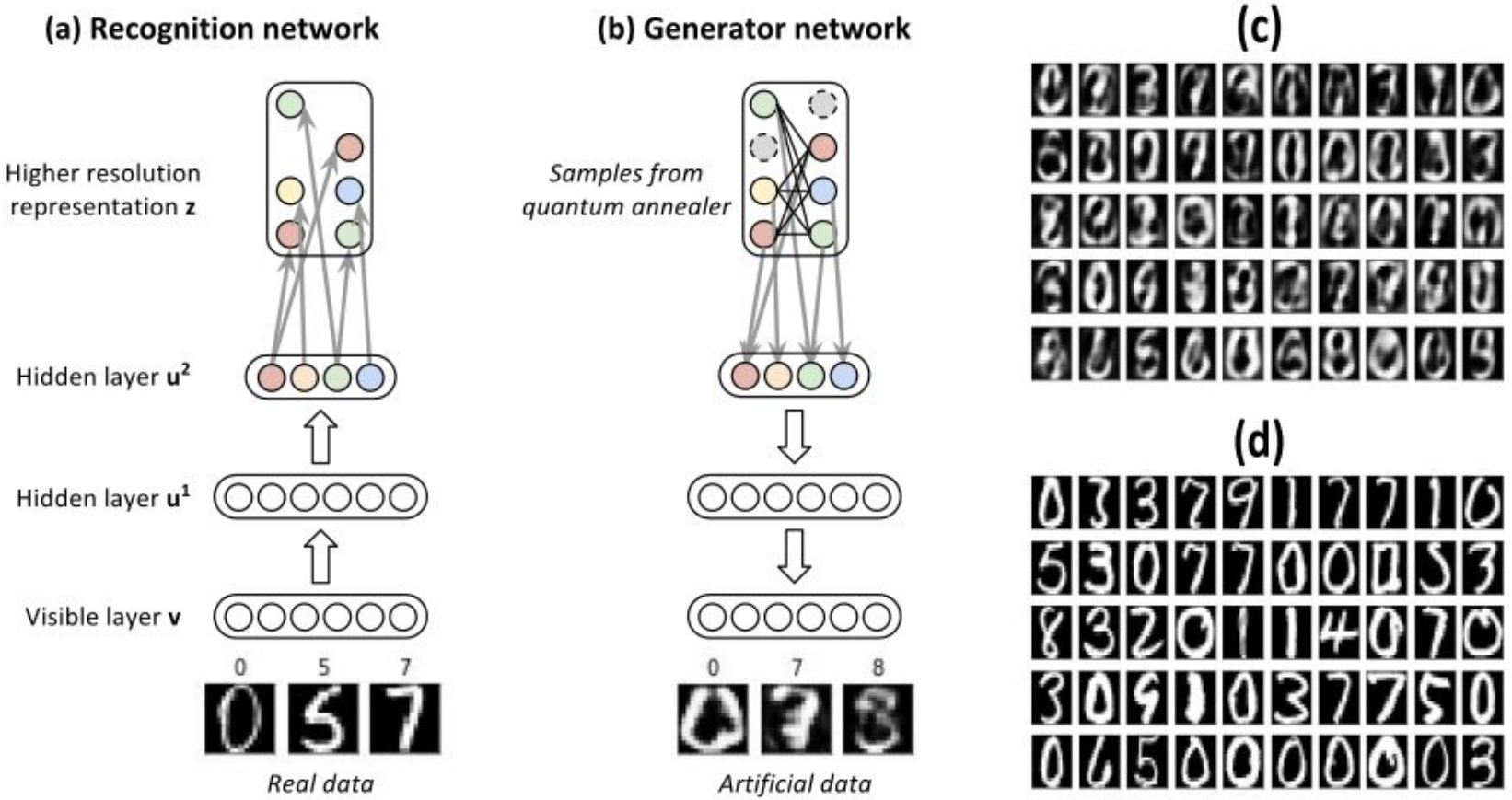


Figure 4 Embedding of RBM visible and hidden nodes onto D-Wave chip layout

Figure retrieved from https://arxiv.org/pdf/1510.06356.pdf

Generative modelling - NASA's results



Experiments using 1644 qubits (no further postprocessing!)

Max. CL = 43

Benedetti, Realpe-Gomez, and Perdomo-Ortiz. Quantum-assisted Helmholtz machines: A quantum-classical deep learning framework for industrial datasets in near-term devices. **arXiv:1708.09784** (2017).

Results - Simulated Quantum Annealing

- Using DWAVE's sampling libraries that run on CPU.

Feature Selection using DWAVE

What is Feature Selection?

- Select the most relevant features in the dataset
- Having additional irrelevant features has two issues:
  - Model learns on these leading to lower accuracy
  - Higher dimensionality leads to higher complexity of the model
- If two relevant features are strongly correlated, discard one of them
- One of the methods to do so: using Mutual Information.

Shannon Entropy

Shannon entropy $H(X)$ quantifies the information in a signal.

$$H(X) = -\sum_{x \in X} p(x) \log p(x)$$

where $p(x)$ represents the probability of an event's occurrence. The less likely the occurrence of an event, the more information is attributed to it. **Conditional Shannon entropy** measures the information in one signal, X, when the value of another signal, Y, is known.

$$\begin{aligned} H(X|Y) &= H(X,Y) - H(Y) \\ &= -\sum_{x \in X} p(x,y) \log p(x,y) - H(Y) \end{aligned}$$

where $H(X,Y)$ measures the information in both signals together, with $p(x,y)$ being their joint probability. Knowing that event X occurs, reduces the information of news that a highly correlated event Y occurs.

Mutual Information

Mutual Information quantifies how much a random variable X knows about Y.

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$
$$= H(Y) - H(Y|X)$$

Conditional mutual information between a variable of interest, X, and a feature, Y, given the selection of another feature, Z, is given by

$$I(X;Y|Z) = H(X|Z) - H(X|Y,Z)$$

where $H(X|Z)$ is the CSE of X conditional on Z and $H(X|Y,Z)$ is the CSE of X conditional on both Y and Z.

Feature Selection

- Maximise $I(X_k; Y)$, the MI between a set of k features, X_k, and variable of interest, Y to select a model's k most relevant of n features. (Hard!)
- Assuming conditional independence of features and limiting CMI calculations to permutations of three features, the optimal set of features is then approximated by:

$$\arg\max_k \sum_{i=1}^{n} \left\{ I(X_i; Y) + \sum_{j \in k|i} I(X_j; Y|X_i) \right\}$$

selects features that best predict the variable of interest

selects features that complement information about the variable of interest

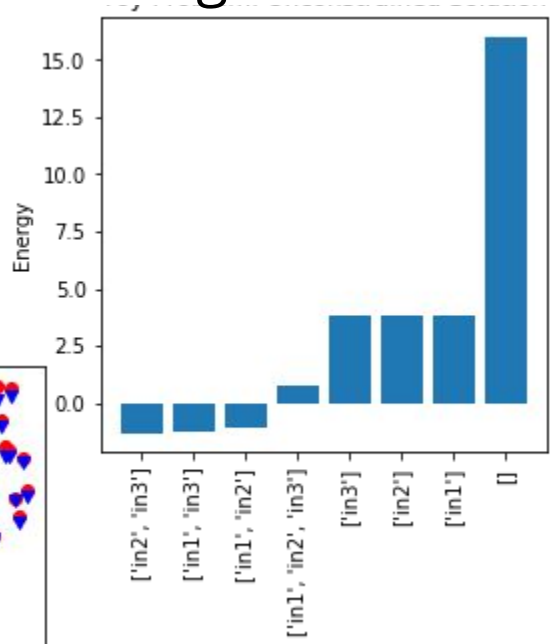
QUBO representation of the problem

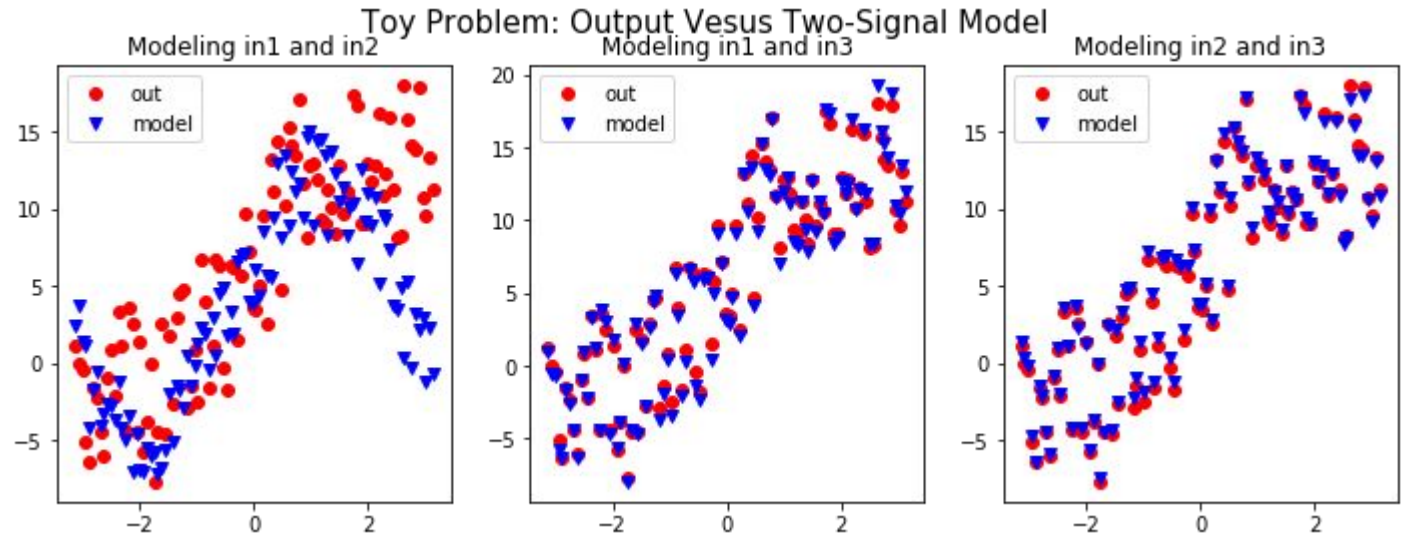
- Represent each choice of $\binom{n}{k}$ features as the value of solution $x_1, ..., x_N$ by encoding $x_i = 1$ if feature X_i should be selected and $x_i = 0$ if not.
- Construct a matrix Q such that : $Q_{ii} = -I(X_i; Y)$ and $Q_{ij} = -I(X_j; Y|X_i)$. (Negative sign converts maximization problem to minimization.)
- QUBO problem can be given as: $\mathbf{x}^T\mathbf{Q}\mathbf{x}$, where $\mathbf{Q}$ is the $n \times n$ matrix and $\mathbf{x}$ is an $n \times 1$ matrix representing the selected features.
- Constraint that exactly k features be selected: penalize solutions that select greater or fewer than k features by adding penalty $P = \alpha \sum_{i=1}^{n}(x_i - k)^2$ to the QUBO. A large enough α can ensure that such solutions are no longer minima of the problem.

Toy Problem

Problem: Three inputs: in1 = sin function, in2 = noisy sin function, in3 = noisy linear function; Output = 2*in1 + 3*in2 + 6*in3

Result: Running on the QPU for k = 2 gives the following energy graph which matches expectation





Questions:

- Why is DWAVE performing so badly on Map Colouring Problem? What preprocessing is required?
- Is QBoost algorithm ***better*** than other classification algorithms, say AdaBoost?
- RBM sampling has been done on simulated quantum annealing. Does using DWAVE Solver API (QPU) give good results?

Further Reading

There is active research going on in Quantum assisted Machine learning. Few other things, researchers have worked upon:

- Bayesian Structure Learning :

  Bryan O'Gorman et. al. ,Bayesian Network Structure Learning Using Quantum Annealing https://arxiv.org/abs/1407.3897

- Support Vector Machines :

  D. Willsch et. al, Support vector machines on the D-Wave quantum annealer, https://arxiv.org/pdf/1906.06283.pdf

References

- DWAVE Documentation
- Steven Adachi and M Henderson , Application of Quantum Annealing to Training of Deep Neural Networks https://arxiv.org/pdf/1510.06356.pdf
- Geoffrey Hinton, University of Toronto, A Practical Guide to Training Restricted Boltzmann Machines, August 2010 https://www.cs.toronto.edu/~hinton/absps/guideTR.pdf
- Biswas et. al, NASA Ames Research Centre, A NASA Perspective on Quantum Computing: Opportunities and Challenges https://arxiv.org/pdf/1704.04836.pdf
- Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers https://arxiv.org/pdf/1708.09757.pdf
- QBoost: Large Scale Classifier Training with Adiabatic Quantum Optimization http://proceedings.mlr.press/v25/neven12/neven12.pdf
- M. Benedetti, Quantum-Assisted Learning of Hardware-Embedded Probabilistic Graphical Models https://journals.aps.org/prx/abstract/10.1103/PhysRevX.7.041052#fulltext
- Effective global approaches for mutual information based feature selection https://dl.acm.org/citation.cfm?id=2623611

To do

DEMO RESULTS OF MAP COLOURING!

Scaling Map colouring(PRORITY) - show graph

RBM running code (TOP PRORITY)

Feature selection make slides (IF TIME)

One slide for BS learning and SVM each(IF TIME)

Show some results :(

Issues Faced

- Incompatibility between qubist and ocean platforms
- DWAVE is shifting platforms, so a lot of documentation is unavailable.
- Downtime as NASA renegotiates contract.
- Libraries not updated to python3

Thank You