Training a Quantum Binary Classifier - Part 1

Problem Formalization

Summary

1. Definition of the Binary Classification problem
2. Loss Function
3. Regularization Term
4. Objective function

Problem Formalization

Strong Classifier

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

$$y \in \{-1, 1\} \qquad h_i : x \mapsto \{-1, 1\}$$

$$x \in \mathbb{R}^M \qquad w_i \in [0, 1]$$

First Training Goal

Minimize the Error over a set of Training Examples

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

Non-Convex $\Rightarrow$ NP-Hard

Second Training Goal

Minimize the complexity of the classifier

Enforce Sparsity

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

All together now

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

NP-Hardness

- Non-convex loss function
- 0-norm regularization

Force inequality constraint

$$y_s \sum_{i=1}^{N} w_i h_i(x_s) \geq 0 \text{ for } s = 1, \ldots, S$$

For every sample

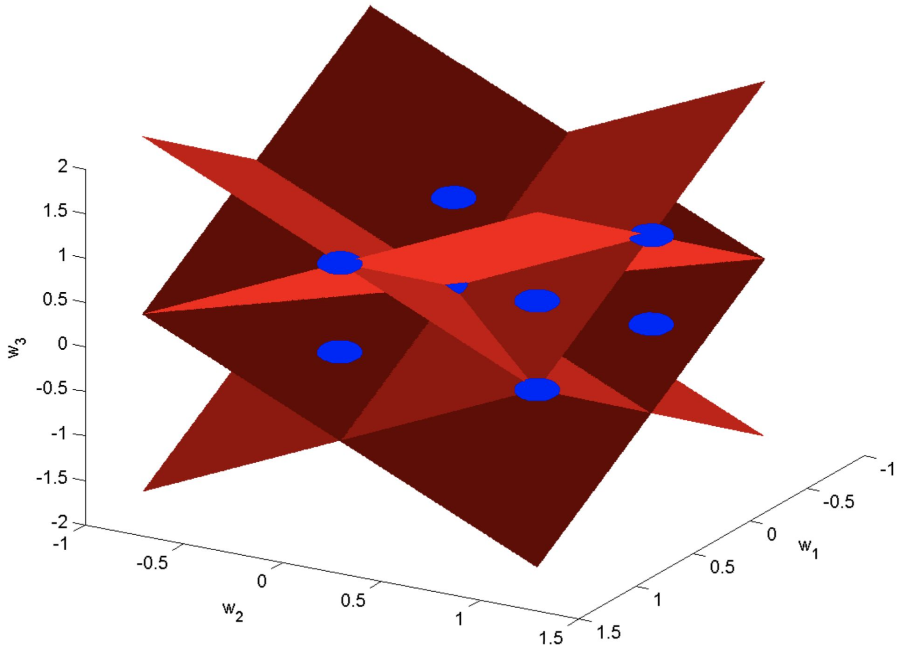
Solution Space

N = 3

$$N_{regions} = (-1)^N \sum_{S_k} (-1)^k (-1)^{dim(\bigcap S_k)}$$

N >= 4

$$N_{regions} \leq \sum_{k=0}^{N} \binom{S}{k}$$



Training a Quantum Binary Classifier - Part 2

Adiabatic version

Summary

1. Training problem for adiabatic quantum computing (AQC)
2. Weight discretization & Bit Precision
3. The D-Wave's AQC loss function
4. The new Objective Function

The Adiabatic formulation for D-Wave

1. Transition from continuous weights to binary variables.
   a. Binary expansion of the weights.
   b. Real valued weights or discrete weights do not influence in the outcome of the classifier if the solution region contains a lattice vertex
   c. The bit precision needed for the weights grows logarithmically with the ratio of the number of training examples to the number of weak classifiers.
2. D-Wave accepts at most quadratic terms
   a. Change the loss to a quadratic loss
   b. Weight finding become Quadratic Unconstrained Binary Optimization problem: QUBO

Weights: From continuous to binary

How many bits do we need? The minimum number possible.

$$\frac{Vertices\ on\ Lattice}{Regions\ in\ Positive\ Quadrant} \approx \frac{(2^{bits})^N}{\frac{N_{regions}}{2^N}}$$

Weights: the math behind

$$\frac{\left(2^{bits}\right)^N}{\frac{N_{regions}}{2^N}} \geq \frac{2^{(bits+1)N}}{\sum_{k=0}^{N}\binom{S}{k}} \geq \frac{2^{(bits+1)N}}{\left(\frac{eS}{N}\right)^N} = \frac{2^{(bits+1)N}N^N}{(eS)^N} \overset{!}{\geq} 1$$

$$\Rightarrow \quad \left(\frac{2^{(bits+1)}N}{eS}\right)^N = \left(\frac{2^{(bits+1)}N}{efN}\right)^N = \left(\frac{2^{(bits+1)}}{ef}\right)^N \overset{!}{\geq} 1$$

$$\Rightarrow \quad bits \geq \log_2(f) + \log_2(e) - 1\,,$$

Aftermath

1. Real valued weights or discrete weights do not influence in the outcome of the classifier if the solution region contains a lattice vertex

2. The bit precision needed for the weights grows logarithmically with the ratio of the number of training examples to the number of weak classifiers.

The new Optimization Objective

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

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

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

$\cancel{\sum y_s^2}$

$$h_i : x \mapsto \{-\tfrac{1}{N}, \tfrac{1}{N}\}$$

The new R(w)

$$R(w) = \sum_{i=1}^{N} \kappa w_i (1 - w_{i,aux}) + \lambda w_{i,aux}$$

$w_{i,aux}$ $\Rightarrow$ Needed if the bit that represents the weights are > 1

References

- https://github.com/dwavesystems/qboost
- https://arxiv.org/pdf/0811.0416.pdf
- https://github.com/CalogeroZarbo/quantum-ml-sagemaker