# Projected Stochastic Gradient Descent with Quantum Annealed Binary Gradients


[1,2]Maximilian Krahn, [3]Michele Sasdeli, [3]Frances Fengyi Yang, [4]Vladislav Golyanik, [1]Juho Kannala, [3]Tat-Jun Chin and [2]Tolga Birdal

[1]Aalto University, Finland [2]Imperial College London [3]University of Adelaide

[4]Max Planck Institute for Informatics


BMVC 2024

## Introduction

Setup:

1) Training BNNs is an NP-hard combinatorial problem

   typical optimiser don't work well on BNNs

2) Using Quantum Annealer to solve this problem

   QA's have the capabilities to solve NP-hard problems

3) Development of an hybrid optimiser

   hybrid algorithms allow for better scalability has current quantum hardware just can solve small problems

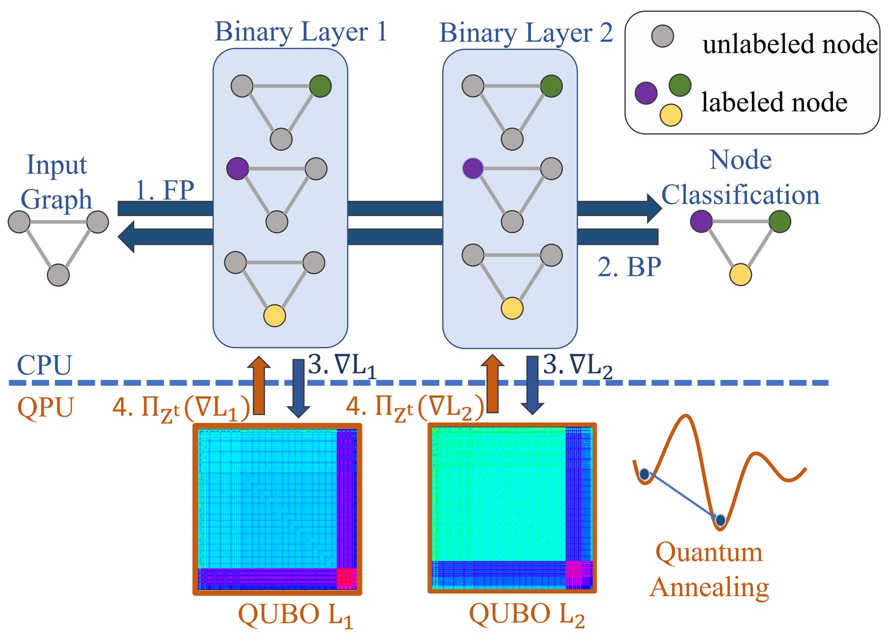



Figure 1. Training Binary Neural Networks with a QA layer by layer

## Contributions

1) We propose QP-SBGD, a novel, **stochastic optimiser** tailored for training binary neural networks utilizing real quantum hardware.
2) We prove that our algorithm **converges to a fixed point** in the binary parameter space under the assumption of the existence of such a point
3) We show an equivalence of our binary projection to a specific QUBO problem, allowing us to implement our algorithm on quantum hardware.

## Quantum Binary Map

We define $\mathbf{\Pi_U} : \mathbb{R}^m \to \{\pm 1\}^n$ to be the map

$$\mathbf{\Pi_U}(\mathbf{v}) := \underset{\mathbf{g}\in\{-1,1\}^n}{\arg\min} \sum_{i=1}^{m} ||v_i - \mathbf{g}^\top \mathbf{u}_i||_2^2. \tag{1}$$

The binary map $\mathbf{\Pi_U}(\mathbf{v})$ in Eq (1) admits the following Ising Model or quadratic unconstrained binary optimisation (QUBO) form:

$$\mathbf{\Pi_U}(\mathbf{v}) = \underset{\mathbf{g}\in\{-1,1\}^n}{\arg\min}\ \mathbf{g}^\top \sum_{i=1}^{m} \mathbf{Q}_i \mathbf{g} + \mathbf{s}^\top \mathbf{g} \tag{2}$$

where

$$\mathbf{s} = -2\sum_{i=1}^{m} v_i \mathbf{u}_i^t, \quad \mathbf{Q}_i = \mathbf{u}_i^t \mathbf{u}_i^{t\top} \quad \text{and} \quad \mathbf{Q} = \sum_{i=1}^{m} \mathbf{Q}_i.$$

## Binary Gradient Approximasation

Let $f : \mathbb{R}^n \to \mathbb{R}^m$ be a differentiable function, $\mathbf{y}$ the output (prediction) and $\hat{\mathbf{y}}$ the target and $E(\mathbf{y}, \hat{\mathbf{y}}) \in \mathbb{R}$ a loss function. We also write: $E_f(\mathbf{x})$ for $E(\mathbf{y}, \hat{\mathbf{y}})$.

$$\mathbf{y} = f(\mathbf{x}), \quad \mathbf{Z}^t \approx \nabla_{\mathbf{y}} E_f \tag{3}$$

When we replace the general matrix $\mathbf{U}$ by a normalized gradient w.r.t. $\mathbf{y}$, namely $\mathbf{Z}^t$ and use $\tilde{\nabla}_{\mathbf{y}} E_f(\mathbf{x})$ as an input our map satisfies the following:

$$\hat{\mathbf{\Pi}}_{\mathbf{Z}^t}(\tilde{\nabla}_{\mathbf{y}} E_f(\mathbf{x})) = \underset{\mathbf{b}\in\mathbb{R}^n}{\arg\min} \|\tilde{\nabla}_{\mathbf{x}} E_f(\mathbf{x})|_{\mathbf{x}^t} - \mathbf{b}\|_2^2. \tag{4}$$

However, our original operator $\mathbf{\Pi}$ projects onto the binary numbers and not the reals. This non-convex map only approximates $\hat{\mathbf{\Pi}}$. Hence, we write:

$$\mathbf{\Pi}_{\mathbf{Z}^t}(\tilde{\nabla}_{\mathbf{y}} E_f|_{\mathbf{x}^t}) \approx \underset{\mathbf{b}\in\{-1,1\}^n}{\arg\min} \|\tilde{\nabla}_{\mathbf{x}} E_f|_{\mathbf{x}^t} - \mathbf{b}\|_2^2. \tag{5}$$

## Binary Update Rule

We now devise a projected variant of SBGD with the distinction that we evaluate the gradients on the variables restricted to $\{\pm 1\}^n$:

$$\begin{aligned} \hat{\mathbf{x}}^t &= \operatorname{sign}(\mathbf{x}^t) \\ \mathbf{x}^{t+1} &= \mathbf{x}^t - \alpha_t \mathbf{\Pi}_{\mathbf{Z}^t}(\tilde{\nabla}_{\mathbf{y}} E_f(\hat{\mathbf{x}}^t)). \end{aligned} \tag{6}$$

We guarantee convergence to a fixed point, if such a point exits.

## Algorithm

Require: Training data $\mathcal{D} = \{(\mathbf{x}_i, \hat{y}_i)\}_{i=1}^{D}$, batch size $B$, learning rate $\alpha$, real initial weights $\{\mathbf{\Omega}^\ell\}_{\ell=0}^{L-1}$

for $t \in [1, \dots, T]$ do
  $\{\mathbf{W}^\ell\}_{\ell=1}^{L} \leftarrow \operatorname{sign}(\mathbf{\Omega}^\ell)$
  Sample a batch index set $\mathcal{B} \subset \{1, \dots, D\}$.
  $\mathbf{y}_{\mathcal{B}} \leftarrow$ Feedforward pass of $\mathbf{x}_{\mathcal{B}}$.
  $\{\dot{\mathbf{r}}_{i,\mathcal{B}}^\ell\}_{\ell=1}^{L} \leftarrow$ Compute intermediate gradients for training data
  for $\ell = 1, \dots, L$ do
    $\dot{\mathbf{W}}^\ell \leftarrow [\mathbf{\Pi}_{\mathbf{Z}_{\mathcal{B},i}^{t,\ell}}(\dot{\mathbf{r}}_{i,\mathcal{B}}^\ell)]_{i=1}^{m}$ By solving the QUBO defined in Eq (2)
    $\mathbf{\Omega}^\ell \leftarrow \mathbf{\Omega}^\ell - \alpha \dot{\mathbf{W}}^\ell$
  end for
end for

We train binary neural networks in a layerwise manner. We start with a forward pass and backpropagation. Then when updating the weights with a quantum annealer / simulated annealer we calculate the binarised weight updates.

## Binary Neural Networks - UCI Adult

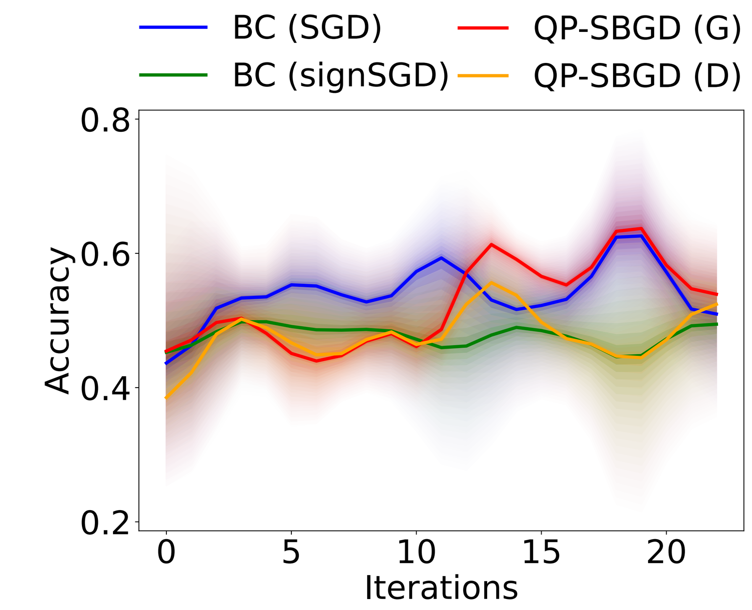



Figure 2. Training accuracy on a subset of the UCI Adult dataset for binary classification.

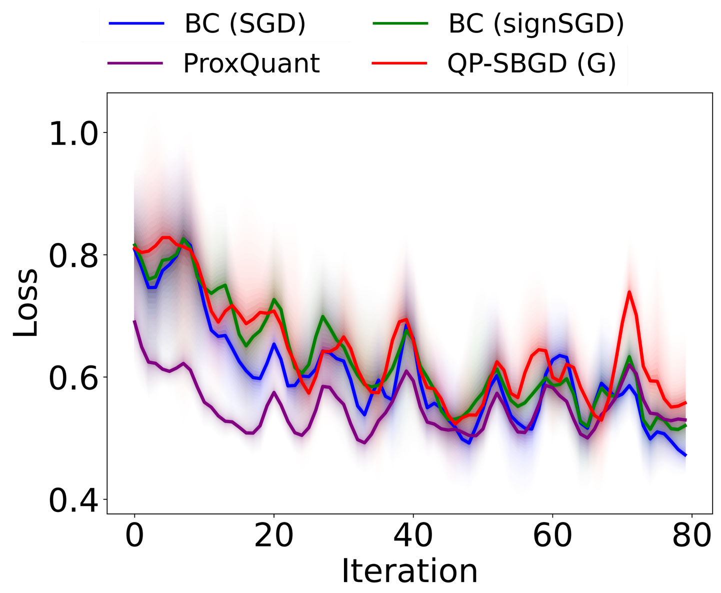



Figure 3. Training loss on a subset of the UCI Adult dataset for binary classification.

## Binary Neural Networks - MNIST Numbers

| | ProxQuant | BC SGD | BC signSGD | QP-SBGD (Gurobi) | QP-SBGD (D-Wave) |
|---|---|---|---|---|---|
| 0/2 | 0.65 | 0.64 | **0.71** | <u>0.66</u> | 0.62 |
| 1/2 | 0.67 | <u>0.72</u> | 0.66 | **0.73** | 0.70 |
| 1/7 | 0.64 | 0.74 | 0.68 | **0.75** | <u>0.74</u> |

Table 1. The accuracy of binary classification on MNIST. The first column contains the digits used in the experiment. We train simple MLPs on 500 training samples and 3000 test samples

## Binary Graph Neural Networks

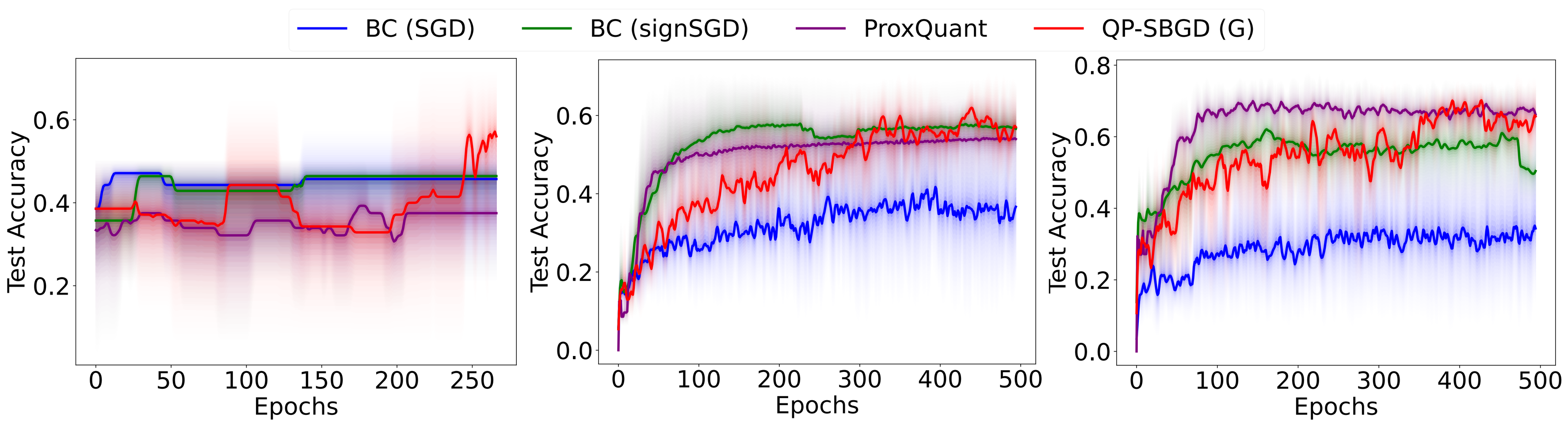



Figure 4. Graph classification: We report mean test accuracy over five runs for Karate club [3] (left), Cora [1] (middle) and Pubmed [2] (right) datasets. of binary GCNs in the node classification task

## QUBO evolution over time

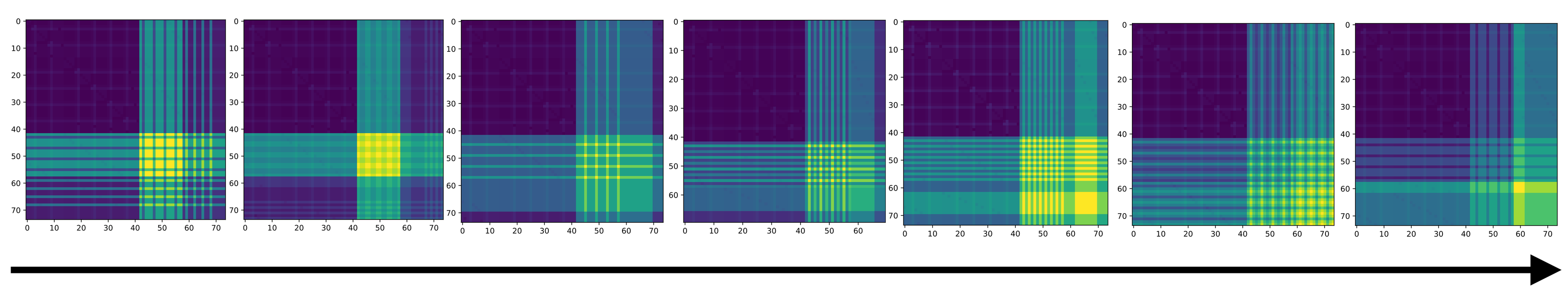
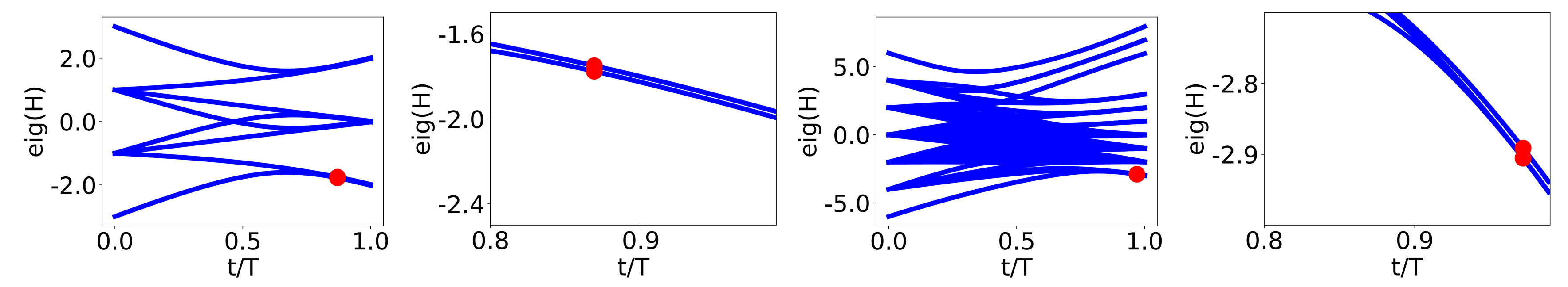


Figure 5. Evolution of the QCBO formulation to calculate the weight updates for the first layer with the Quantum Projected Stochastic Binary-Gradient Descent algorithm.

## Hamiltonian Analysis




(a) $N_\ell = 1$ neuron (b) Close-up of (a) (c) $N_\ell = 2$ neurons (d) Close-up of (c)

Figure 6. The eigenvalues of the Hamiltonian of the QUBO in Eq (2) while training a binary MLP on the adult dataset. Those eigenvalues are plotted as a function of annealing time $(t/T)$ for a linear layer with one to four neurons and batch size 1. The red bar represents the eigenvalue gap between the ground level and the first excited level that does not evolve into the ground state.

## References


[1] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning.

[2] Galileo Namata, Ben London, Lise Getoor, Bert Huang, and U Edu. Query-driven active surveying for collective classification. In 10th international workshop on mining and learning with graphs, 2012.

[3] Wayne W Zachary. An information flow model for conflict and fission in small groups. Journal of anthropological research, 1977.


## Contact and Website


Email: maximilian.krahn@icloud.com
Website: `https://qpsbgd.github.io`