Quantum learning pipeline

Exploring an implementation for Support Vector Machines

Mario Bifulco, Luca Roversi

10th September

University of Turin

Why Quantum Support Vector Machines?

Classical SVM limitations

- Quadratic complexity scaling
- Kernel calculation bottlenecks
- Hard optimization for high-dimensional data

Quantum Computing Advantage

- Quantum parallelism for kernel calculation
- Quantum tunneling to escape local minima

What is a Quantum Support Vector Machine?

Gate-based approach

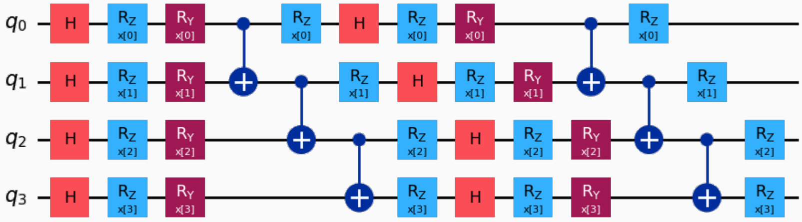
1. Encoding classical data via quantum feature map
2. Estimate kernel via quantum circuit evaluation
3. Classical SVM training with quantum kernel

Annealing-based approach

- Classical kernel calculation
- SVM optimization problem reduction to QUBO formulation
- SVM training via quantum annealing

Quantum kernels

1. Raw input encoding
2. PCA to retain N most significant features (8, 16, 30)
3. Encoding with a feature map ($\mathtt{Z}$, $\mathtt{ZZ}$, $\mathtt{SU2HR}$, $\mathtt{SU2RR}$) repeated k times (1, 2, 3)
4. Compute kernel as $|\langle\phi(x_i)|\phi(x_j)\rangle|^2$



Finding a good quantum kernel

# Qubits	Feature Map	Repetitions	Alignment (KTA)
30	SU2HR	1	98.650%
8	ZMAP	2	93.558%
16	SU2RR	1	91.203%

Quantum annealing SVM

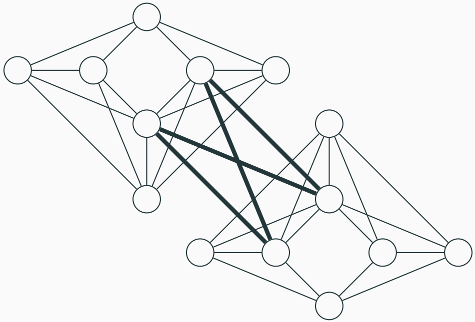


Figure 1: 14 qubit QPU

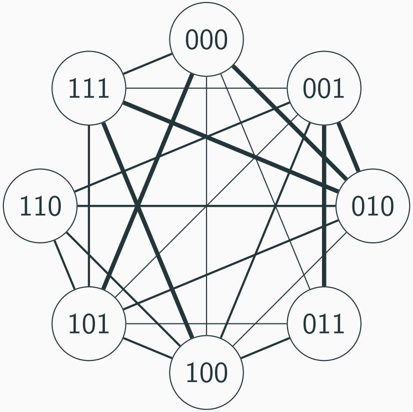


Figure 2: States of a 3 bit problem

Choosing an appropriate regularization parameter

Problem

SVM requires a fixed regularization parameter (C)

This parameter must be represented in binary format

Solution

Set the upper bound of C to $2^n - 1$ to minimize the number of unused bits

Fully quantum pipeline

Training

1. **Gate-based kernel calculation**
2. *Kernel matrix export*
3. **Annealing-based optimization**

Output

A model ready for classical inference

Results

Main findings

We achieved an `F1-score` of 90%

- 30 qubit kernel
- an upper bound of 255 for the regularization parameter

This model performs comparably to a classical SVM with an RBF kernel

Future works

- Improved preprocessing pipeline
- Comprehensive analysis of the impact of kernel dimensionality
- Try different encodings for C to map a real number

Thanks for your attention!

For further questions, feel free to contact me (`mario.bifulco@unito.it`)

Access the code and read the full paper