Applications of Quantum Computing in Time Series Analysis

Nishad Singh, Pranay Patil, Nathaniel Parke, Matthew Anderson

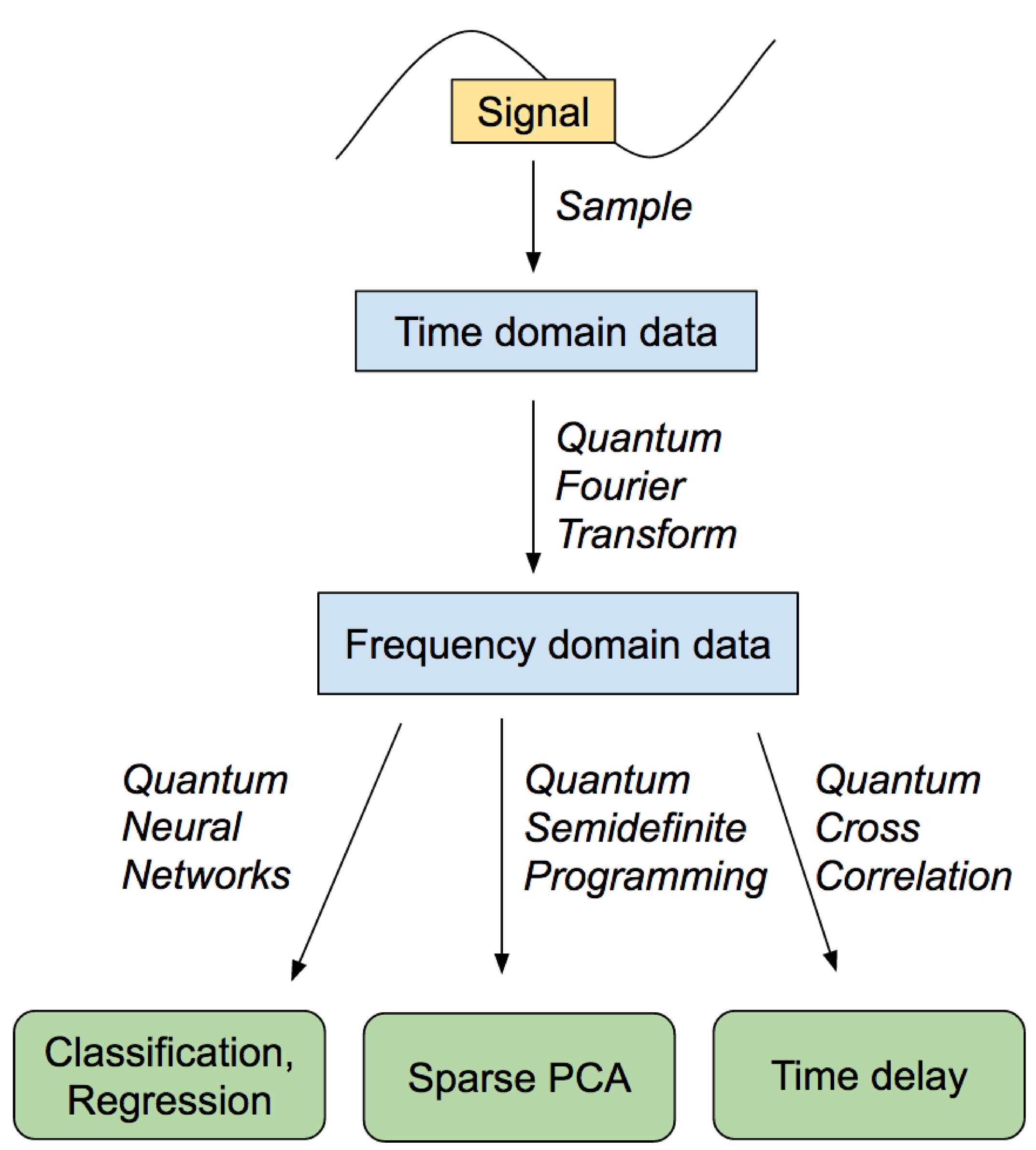
Department of Computer Science - University of California, Berkeley

Objective

A time series is a sequence of numerical data points in successive order. Time series data appears in countless domains, from financial markets to traffic engineering. Analyzing time-series data quickly can be incredibly valuable.

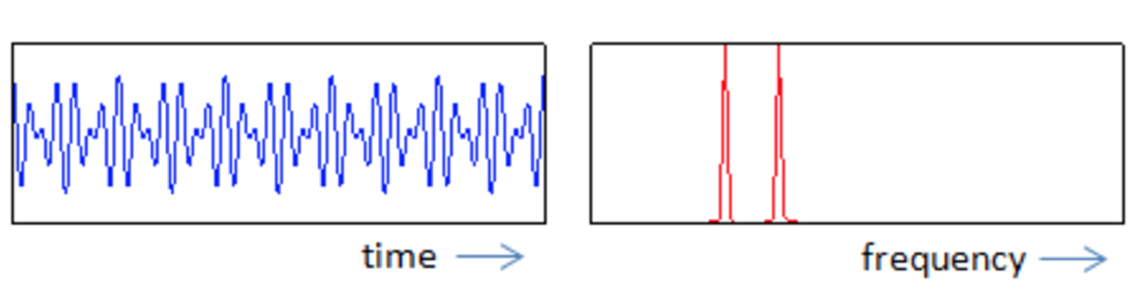
Our objective is to explore some common techniques used in time series analysis and identify areas in which quantum algorithms can provide meaningful improvements.

Overview



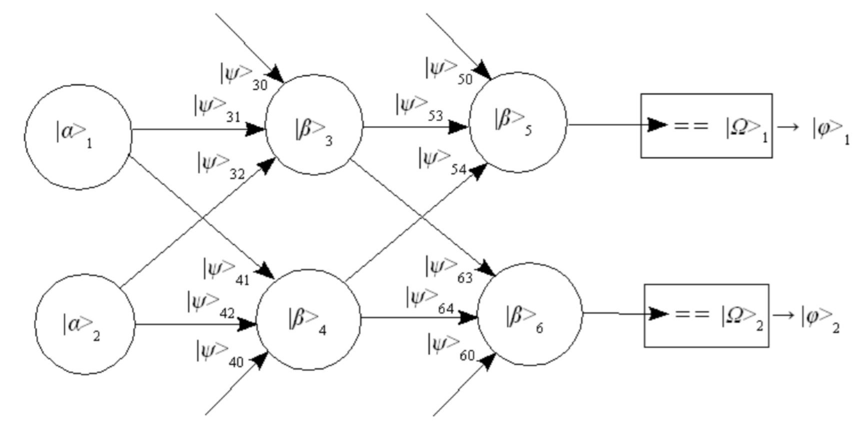
Frequency domain

Time series data is often periodic. Frequency-domain representations can be more concise and make accessible more meaningful features than time-domain representations of this data.



Quantum Neural Networks

It is desirable to find a quantum approach to training **artificial neural networks**. Consider the XOR neural network below.



At each node, the output of the previous layer is combined and multiplied with a weight vector and then summed, to produce a weighted linear combination. The desired output of training are the weights $|\Psi\rangle_{i,j}$ for all nodes in the network. Classical approaches learn these weights through sequential iteration of the back-propagation algorithm which finds the **gradient of the loss function** and steps stochastically towards the minimum.

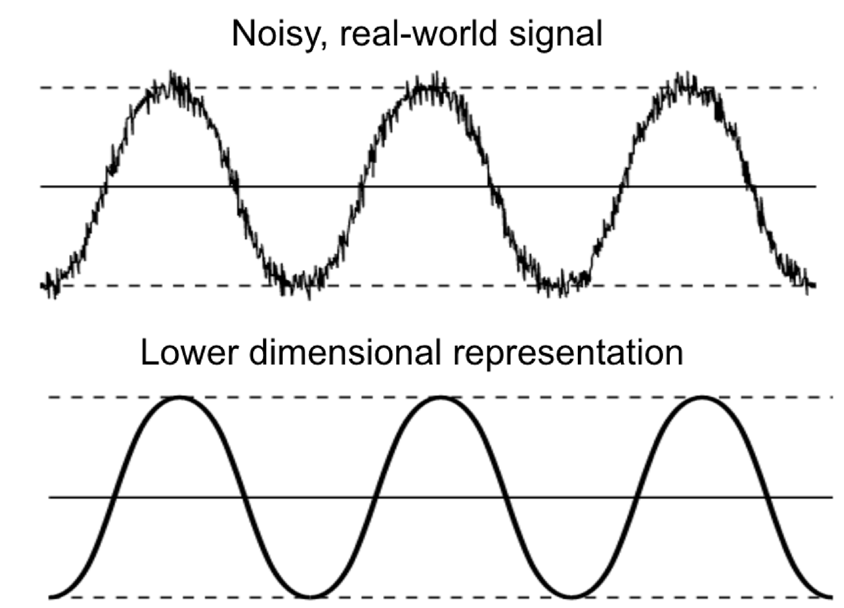
Through the use of **Grover's algorithm**, quantum neural networks can be trained in a different manner, by initializing a random set of weights for all nodes, then running Grover's algorithm to find the optimal weight vector for a particular node, holding all other node weights constant. It is unclear whether this method provides speedup over classical back-propagation algorithms. **Complexity increases exponentially** in the number of connections to a particular node and the number of bits required to represent weight values. This is a randomized algorithm, runtime results will vary.

QNN Takeaways

- Using Grover's search to update specific weight vectors at a particular node, could provide advantages in Artificial Neural Networks
- Search space increases exponentially with connectedness and precision of weight values
- Runtime comparisons are unclear, and need future work

Quantum Semidefinite Programming

When analyzing time series data, it is important to be able to deconstruct it into a few **principal components** that can capture a majority of the variance. In other words, we want to derive a sparse PCA formulation of the input data that is able to erase most of the noise.



Currently, the best known way to do so is by solving a **semidefinite programming** (SDP) optimization problem, which takes polynomial time.

A quantum approach to SDP (QSDP), however, provides a quadratic time speedup in the general case. It does so by transforming the optimization problem into a **feasibility problem** that, in every iteration, outputs any solution satisfying the constraints below a given threshold t. QSDP then performs a binary search to find the minimum threshold for which a solution exists which then becomes the solution to the optimization problem.

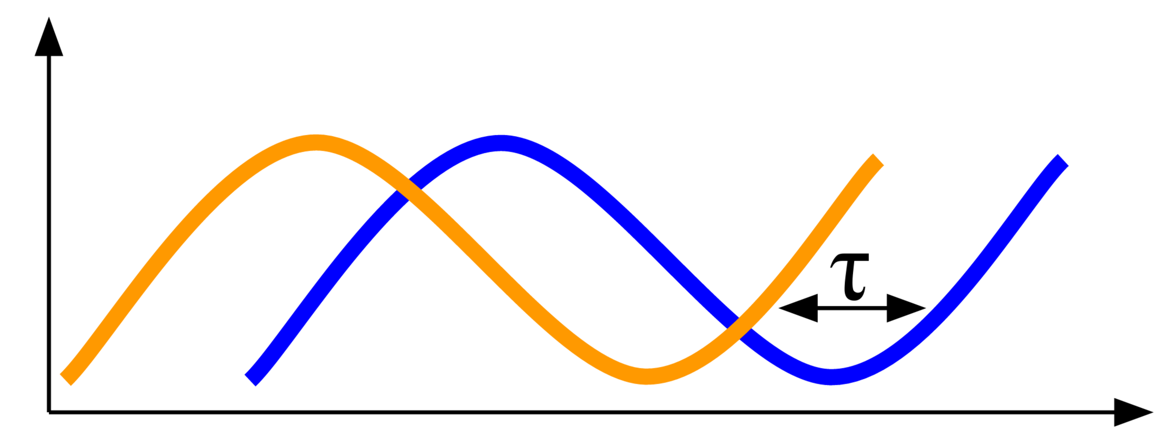
The main quantum advantage comes from the preparation of **quantum Gibbs states** which allow the algorithm to obtain samples more efficiently.

QSP Takeaways

- Sparse PCA formulations remove noise from time series data.
- SDP can be used to generate sparse PCA formulations of input data.
- QSDP provides a quadratic time speedup over classical SDP by turning the problem into a feasibility one and using quantum Gibbs states.

Quantum Cross Correlation

In time series analysis, often valuable to find the **time-delay** between two signals (say, to predict the value of a trailing stock).



One can do this by finding the time shift under which their **cross-correlation** is maximized. Suppose we sample the two signals f and g at T evenly spaced intervals, giving vectors $\vec{f}$ and $\vec{g}$. Computing their cross correlation vector in the time domain would require solving

$$(f \vec{*} g)[t] = \sum_{m=0}^{T-1} (f[m])^* g[m+t]$$

for all $t \in \{0, 1, ..., T-1\}$.
However, if we have the frequency-domain representation of $\vec{f}$ and $\vec{g}$, $\mathcal{F}\{f\}$ and $\mathcal{F}\{g\}$, by the **Cross Correlation Theorem**, we can compute our cross correlation with

$$f \vec{*} g = \mathcal{F}^{-1}\{(\mathcal{F}\{f\})^* \mathcal{F}\{g\}\}$$

To find the time delay, we compute

$$\tau = \arg\max_{t \in \{0,1,...,T-1\}} (f \vec{*} g[t])$$

We can compute the all the Fourier transforms using the **QFT**, the Inverse Fourier transform with the IQFT, and take the argmax using **Grover's algorithm** with an oracle that tags elements above a variable value. This would take $O(T)$ time instead of the classical $O(Tlog(T))$ time.

QCC Takeaways

- Time delays between signals are common in time series data (say, trailing stocks)
- Identifying them quickly can be very useful
- Quantum techniques can provide asymptotic speedups from classical approaches