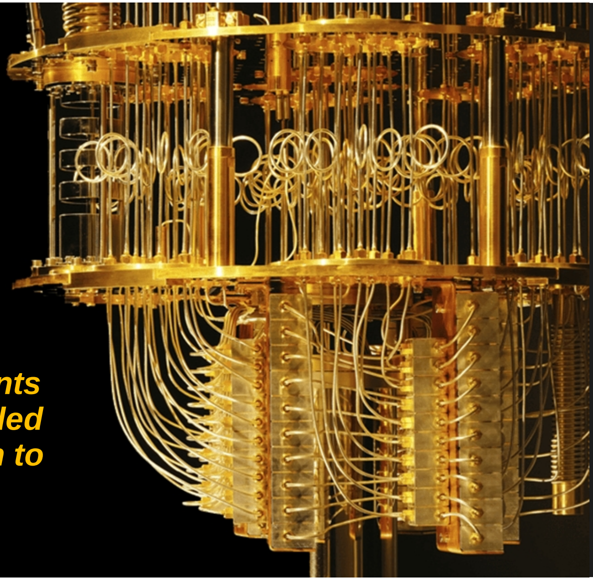
The aims of this session:

To provide workshop participants with knowledge and skills needed to engineer a quantum solution to practical time series analysis problems



Part 1 – Introduction and fundamentals
Key concepts in classical time series
Quantum time series analysis and forecasting
QTSA data encoding and analysis
QTSA with variational quantum linear regression
Break

Part 2 – QNN inspired solutions
QTSA with variational quantum Fourier transforms
QTSA with quantum neural networks
QTSA for multi-variate time series
Summary and reflection

Time Series Analysis Using QML, Part 1
Introduction and Fundamentals

Jacob L. Cybulski
School of IT, SEBE, Deakin University

13 October 2022

Session Aims and Objectives

- There are many quantum algorithms useful in finding solutions to a practical data-intensive problem, such as time series analysis and forecasting.

  So…

- Can we engineer a quantum solution to such a data-intensive problem?
- Yes, we will do so here! We will introduce Quantum Time Series Analysis (QTSA)
- We aim for the stars, but…

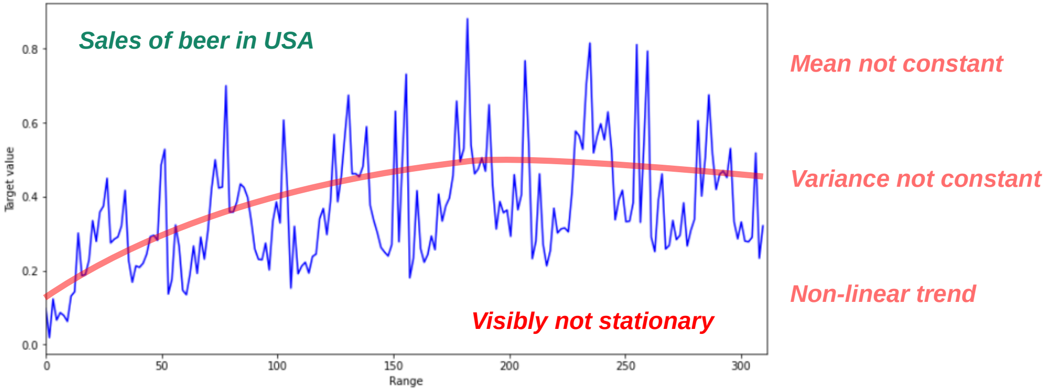
- In this workshop participants will learn QTSA fundamentals - how to:
  - *Understand* properties of complex data, such as time series
  - *Encode* such complex data into a quantum circuit
  - *Process* data encoded in a quantum circuit
  - *Interpret* the outcome of the circuit execution
  - *Parameterise and train* a quantum circuit given a data set

- *We will not seek to explore quantum advantage of QTSA solutions over classical ones but rather aim to gain experience in quantum manipulation and modelling of time series data.*

- *Some prerequisites knowledge needed:*
  - *Variational quantum circuits concepts*
  - *Quantum neural networks concepts*
  - *Exposure to Python and Qiskit*

Key concepts in time series analysis

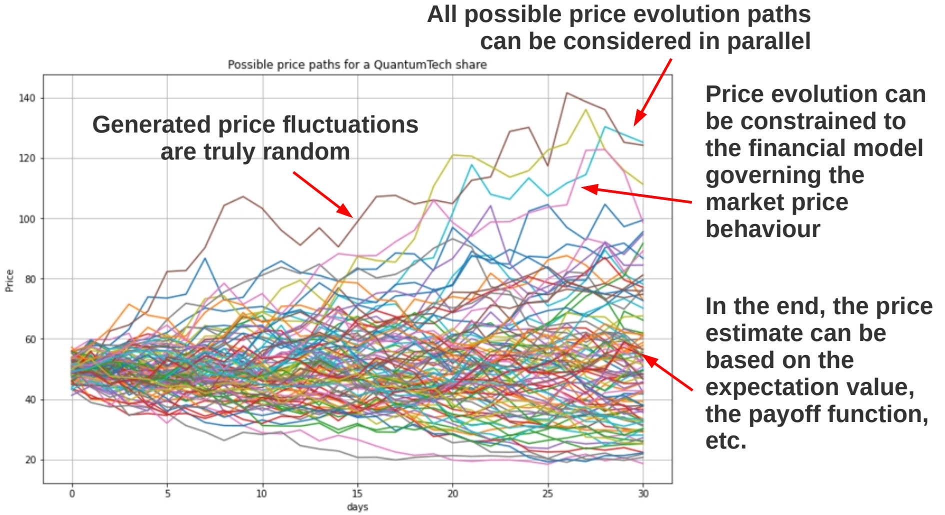
- Time series analysis aims to *identify patterns* in historical time data and to *create forecasts* of what data is likely to be collected in the future
- Applications include heart monitoring, weather forecasts, machine condition monitoring, etc.
- Times series analysis is well established with *excellent tools* and *efficient methods*, yet some organisations aim to improve them further
- Time series must have an *unique index* - a time-stamp sequencing the series
- Time series needs to be *ordered* by its index
- Time series will also have some *time-dependent attributes* to be modelled
- Time series can be *univariate* or *multivariate*, depending on whether a single or multiple attributes are being investigated
- *Missing indeces* and their dependent attributes may need to be imputed (e.g. interpolated)
- Index needs to be of appropriate *granularity*, e.g. years, months, weeks, days, hours, etc.
- Attributes need to be *aggregated* to the required index granularity
- Time signal often shows *seasonality* in data, i.e. a regular repeating pattern
- With aggregation and smoothing seasonality can be removed and *trends* visually identified
- Majority of forecasting methods require *time-series to be stationary*, i.e. its mean, variance and auto-correlation are constant
- *Quantum time series analysis (QTSA) is a promising approach to time series analysis and forecasting!*



Quantum Computing
In Time Series Analysis

- Quantum computing helps solving problems in many disciplines, e.g.
  - *natural science*, such as calculation of molecular energy or protein folding;
  - *finance*, such as portfolio optimisation, pricing of financial options or credit risk assessment;
  - *optimisation*, such as in vehicle routing or energy distribution using quantum-enhanced optimisation techniques;
  - *machine learning*, featuring many general purpose algorithms, such as neural networks or kernel methods.
- However...
  What problems fit quantum solutions?
- What time series analysis applications could benefit from quantum technology?

- Quantum applications can demonstrate their advantage over classical solutions by relying on the following features of quantum systems:
  - Randomness of observable results *(measurement of quantum states)*
  - Pursuing alternative decisions concurrently *(superposition of quantum states)*
  - Controlling parallel choices with constraints *(entanglement of quantum systems)*
- QTSA example where all three principles have been applied – financial option price prediction

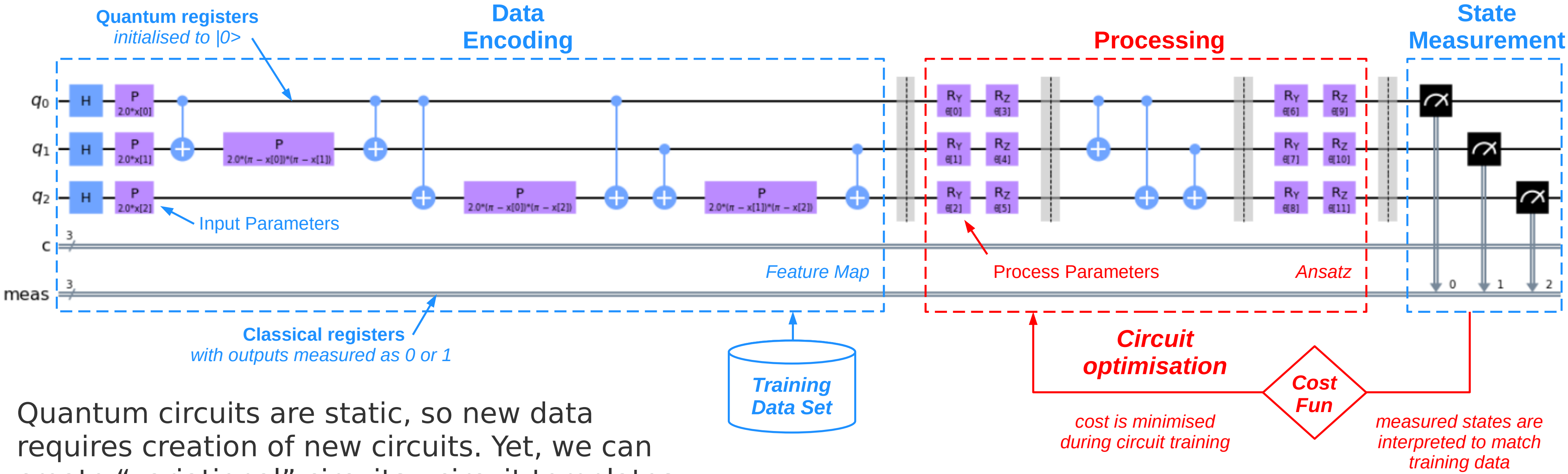


Practice 1
Preparation and orienteering

Plan
- Show access to GitHub ironfrown
- Show access to IBMQ Labs
- Show access to jacobcybulski.com
- Demonstrate IBMQ Labs setup

- Create an IBMQ account and login — URL: https://quantum-computing.ibm.com/
- Access ironfrown code: — URL: https://github.com/ironfrown/qtsa_workshop
- Download workshop files: — Code → Download ZIP
- Upload files to IBMQ lab: — Follow ironfrown GitHub repo Readme file
- Run and explain utils: — Follow live / recorded session and demos

Variational Quantum Circuits
Quantum problem solving



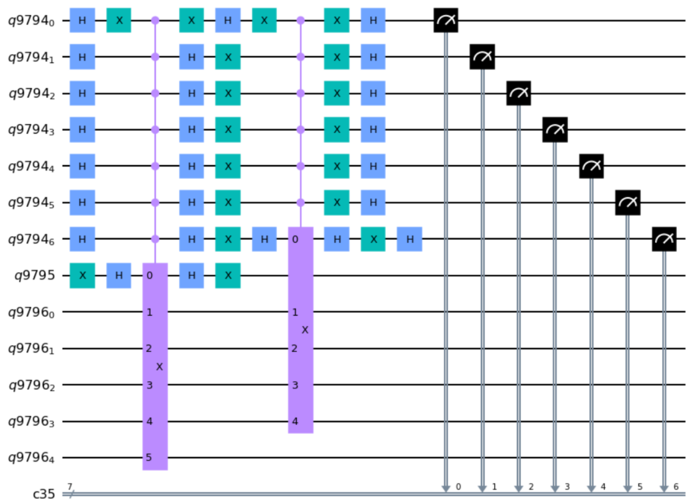
Quantum circuits are static, so new data requires creation of new circuits. Yet, we can create “variational” circuits - circuit templates with parametrised gates – e.g. P, Rx, Ry and Rz gates with varying degrees of rotation, which can be optimised using some ML algorithm.

To train a variational circuit, its parameters are repeatedly instantiated, the circuit executed, the states of its qubits measured and returned.

The measurement outcomes can be re-interpreted into a desired form (binary, integer or float). Such results are compared against the expected values (using training data) and used by the cost function to guide the selection of new values for the process parameters, which are then combined with new input parameters to form the next version of the circuit, and improving performance.

In what order?

At 1966 Cannes Film Festival the following conversation was observed between two famous film directors: Henri-Georges Clouzot and Jean-Luc Godard.



Clouzot: But surely you agree, M. Godard, that films should have a beginning, a middle part and an end?

Godard: Oui,
but not necessarily in that order!

Albert Einstein: But surely you agree, Herr Bohr, every quantum circuit should have data encoding, data processing and measurement of its states?

Niels Bohr: Ja,
but not necessarily in that order!

Albert Einstein (mumbles): Hmmm, God does not play dice (ahem)...

Wikipedia: https://en.wikipedia.org/wiki/File:Contact_print.jpg

Ways of encoding time series data for the quantum system processing

Note similarities with measurement interpretation!

- *Quantum systems have no memory!*
- *Quantum circuits take no data!*

- The only way of obtaining and retaining information in a quantum system is via:
  - *structure of a quantum circuit* (as done in variational methods)
  - *states of quantum computation* (as done in adiabatic optimisation)
- In this presentation we will focus on the first option – the *variational methods*
- In variational time series analysis, the key concerns are:
  - *data encoding strategy*
  - *circuit optimisation strategy*

There are many different quantum data encoding / state preparation methods:

- *basis encoding*, with qubits acting as bits in the encoded number (logical / integer) to be processed later in the circuit
- *angle encoding*, where qubit rotation (float) represents the value of data
- *amplitude encoding*, where each data point is encoded as expectation value of circuit measurement (float), usually no further data processing is present
- Others: QuAM, QRAM, Qsample, ...

We will explain these approaches by QTSA demonstration of:

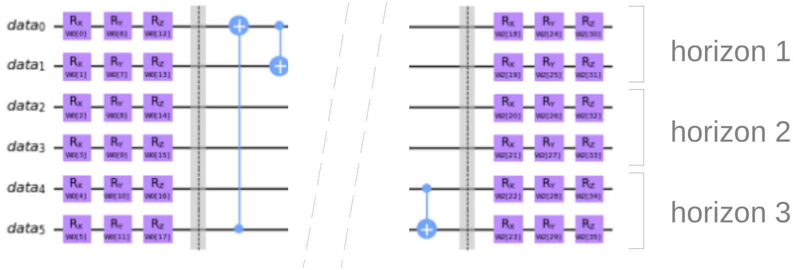
- *Quantum regression (function fitting)*
- *Quantum Fourier transform (function fitting)*
- *Quantum neural networks (pattern detect.)*

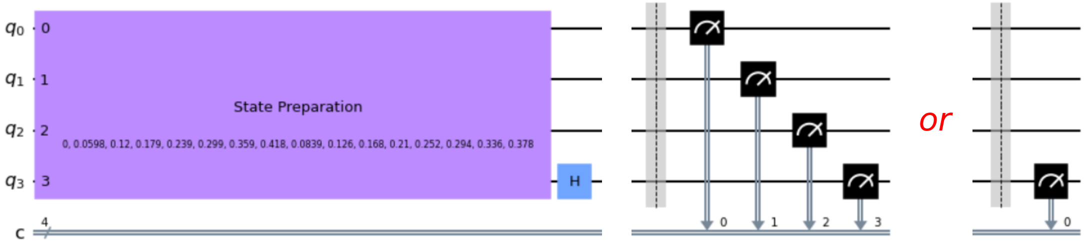
Schuld, Maria, and Francesco Petruccione. *Machine Learning with Quantum Computers. 2nd ed.* 2021 edition. Springer, 2021. https://link.springer.com/book/10.1007/978-3-030-83098-4.

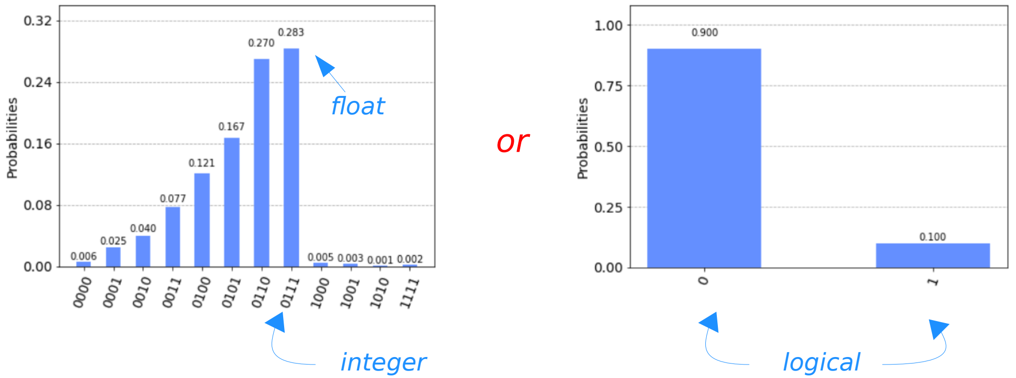
Ways of Measuring Outcomes

There are many ways of obtaining the outcome of a circuit execution.

- We can select all qubits to measure
- We can select only those qubits that give you (theoretically) the most appropriate result
- We can interpret the counts of multiple measurements
- We can reinterpret circuit measurements into different combinations of outcomes, e.g. to predict larger QTSA horizons







Repeated circuit measurement can be interpreted as outcomes of different types, e.g.

- as a binary outcome
(e.g. a single qubit measurement),
- as a bitwise representation of an integer number
(e.g. most frequent combination of multi-qubit measurements), or
- as a value of a continuous variable
(e.g. expectation value of a specific outcome).

Putting it all into practice

Variational quantum linear regression and polynomial fitting

The variational linear regression methods can be extended to polynomial fitting

"Not necessarily in that order"

Possibly in an unexpected part of a quantum solution

We are trying to find *a* and *b* to satisfy a linear equation:

$$\overrightarrow{y} = a\overrightarrow{x} + b$$

We will encode a normalised vector *y* as a quantum state $|y\rangle$

$$|y\rangle = \frac{1}{C_y}\overrightarrow{y} = \frac{1}{C_y}(a\overrightarrow{x} + b)$$

We will identify *a* and *b* in the optimisation process, which considers a sequence of states:

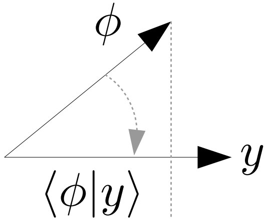
$$|\phi\rangle = \frac{1}{C_\phi}(a\overrightarrow{x} + b)$$

$$|\phi_0\rangle, |\phi_1\rangle, ..., |\phi_n\rangle$$

Starting with $|\phi_0\rangle = |\phi\rangle\,(a_0, b_0)$

The optimisation will search for such *a* and *b*, and thus $|\phi\rangle$ to minimise the cost function C_p which tries to maximise the similarity of $|y\rangle$ and $|\phi\rangle$

$$C_p(\phi, y) = (1 - \langle\phi|y\rangle)^2$$



What's remaining is to create a quantum circuit able to calculate $\langle\phi|y\rangle$ based on pairs *a* and *b*, so that the cost function could drive the optimisation process

The required circuit can be built into the cost function.

It will rely on the *amplitude encoding* of sample data, which ensures that measured expectation values of the composite qubit states corresponds to data point values

We encode normalised *ax+b* into half of the qubits, and normalised *y* into the remaining ones. These encoded values will constrain possible circuit states.

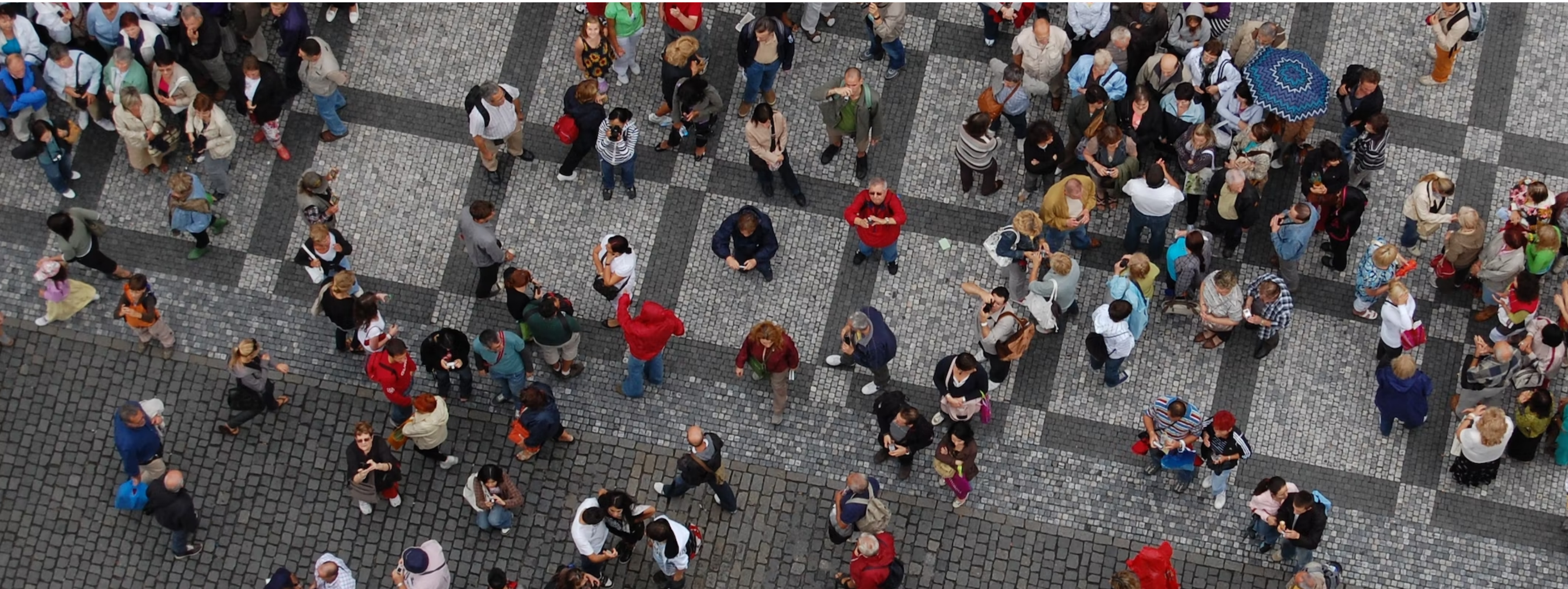
The Hadamard gate *H* allows measuring the expectation value, which implies the value of the inner product of interest *(for more details, see ref: Qiskit 2020)*

Bravo-Prieto, Carlos, Ryan LaRose, Marco Cerezo, Yigit Subasi, Lukasz Cincio, and Patrick J. Coles. "Variational Quantum Linear Solver." *ArXiv Preprint* ArXiv:1909.05820, 2019.

Qiskit. "Variational Quantum Regression", in *Learn Quantum Computation Using Qiskit. Textbook*, 2020. https://qiskit.org/textbook/ch-demos/variational-quantum-regression.html

Practice 2
Linear regression (Demo)

Plan
- Navigate IBMQ Labs
- Explain there are utilities (for self-exploration)
- Explain there is linreg demo (for self-exploration)
- Show results on the next page



Experiments R1-5

Linear Regression

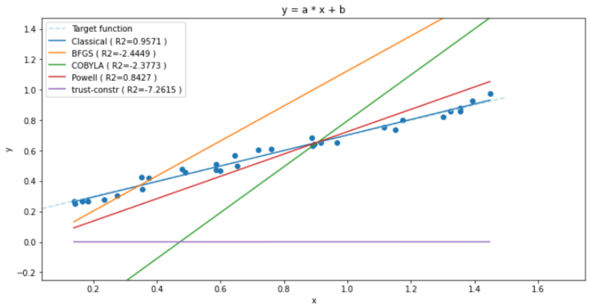
Linear regression over purely linear data.

Data is only 32 points. Fitting was very slow!

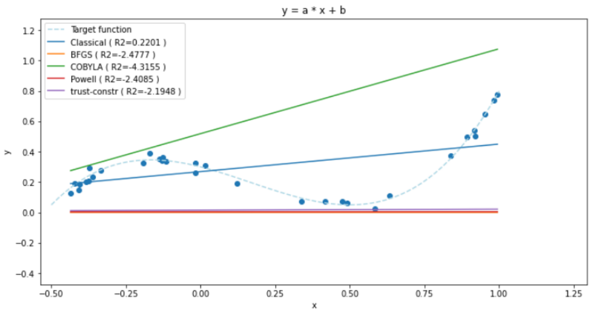
Only one optimiser gave good fit (Powell).

Different optimisers give vastly different results.

Initial values to the optimisation had huge impact on the result.



f(x): 0.5x + 0.2



f(x): $0.3-0.5x-x^2+2x^3$

Linear regression over polynomial data.

Data is only 32 points. Here training data is only 32 points.

Data fitting was extremely slow (esp. Nelder-Mead).

When initial values were close to zero, optimisers were giving good results.

With the exception of a classical solution all fits are slightly offset!

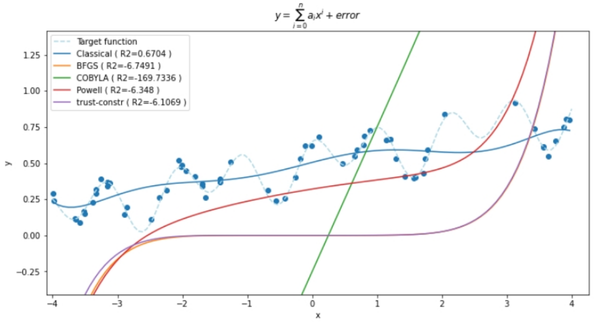
Polynomial Fit

Polynomial fit of 7th order over complex trigonometric data with trend.

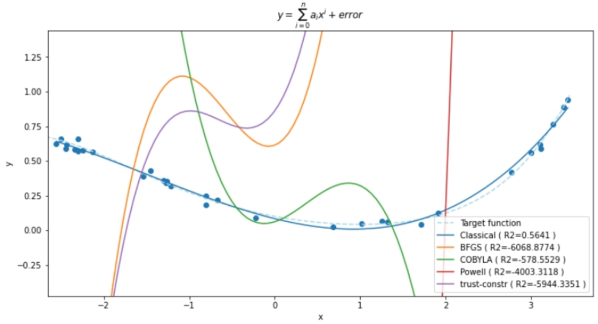
Data was 32 points.

Selection of initial coefficients was crucial.

As a result of good starting point (close to zero), the fit was quite good, yet not optimal, regardless of which optimiser was in use.



f(x): 0.5+0.09x+0.09sin(3x)+0.15cos(6x)



f(x): $-(8x-4x^2+0.2x^3-0.1x^5)/70+0.1$

Polynomial fit of 7th order over 5th order polynomial.

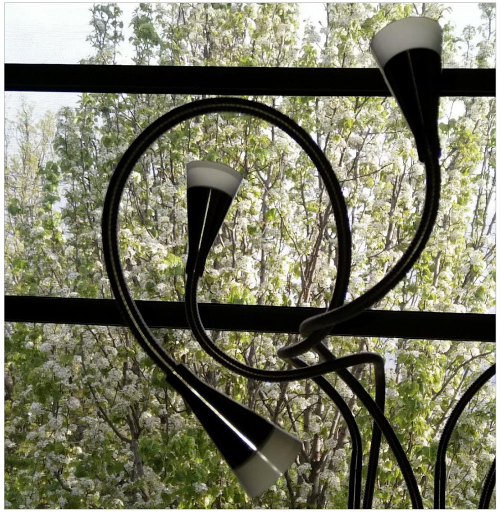
Data was 32 points.

Data fitting was very slow.

As the initial coefficients were random, none of the optimisers could fit the curve.

Variational quantum linear regression Reflections

- Variational quantum regression can only be used to fit linear data (with some noise)
- Its training convergence is highly sensitive to the optimisation strategy, optimisers’ hyper-parameters, and their initialisation
- While the approach taken can be easily adapted to fitting higher order polynomials, only certain types of functions fit successfully (mainly those appearing in publications)
- Potentially, barren plateaus affect optimisation
- As compared with classical methods of linear/polynomial fitting, experiments with variational quantum regression indicate the adopted quantum regression approach is not promising
- We will therefore seek other approaches to time series analysis, its data fitting and forecasting!

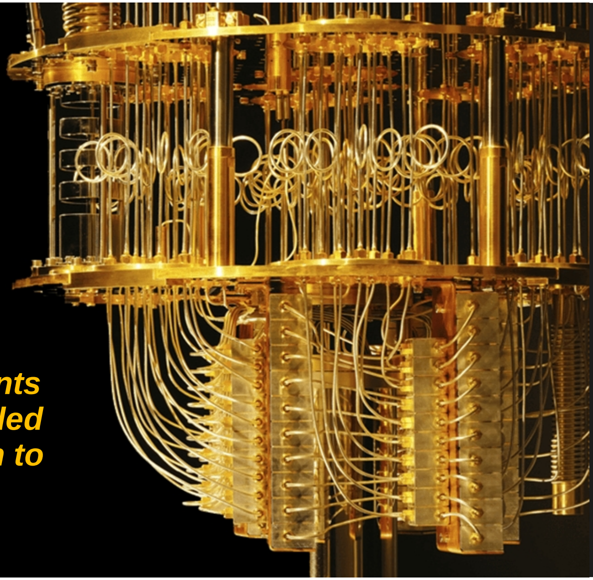


It is worth noting that other, more recent methods such as QSVT (Quantum Singular Value Decomposition), can assist fitting any function (or data) with higher-order polynomials

End of Part 1

The aims of this session:

To provide workshop participants with knowledge and skills needed to engineer a quantum solution to practical time series analysis problems



Part 1 – Introduction and fundamentals
Key concepts in classical time series
Quantum time series analysis and forecasting
QTSA data encoding and analysis
QTSA with variational quantum linear regression
Break

Part 2 – QNN inspired solutions
QTSA with variational quantum Fourier transforms
QTSA with quantum neural networks
QTSA for multi-variate time series
Summary and reflection

Time Series Analysis Using QML, Part 2

QNN Inspired Solutions

Jacob L. Cybulski

School of IT, SEBE, Deakin University

13 October 2022

Variational quantum Fourier transforms

Schuld, Maria, Ryan Sweke, and Johannes Jakob Meyer. "The Effect of Data Encoding on the Expressive Power of Variational Quantum Machine Learning Models." *Physical Review A* 103, no. 3 (March 24, 2021)

Pérez-Salinas, Adrián, Alba Cervera-Lierta, Elies Gil-Fuster, and José I. Latorre. "Data Re-Uploading for a Universal Quantum Classifier." Quantum 4 (Feb 6, 2020): 226.

PennyLane. "Quantum models as Fourier series", 2021. https://pennylane.ai/qml/demos/tutorial_expressivity_fourier_series.html

We consider a quantum model, which takes a vector of 1D data.

The model f_θ consists of *n* layers, each with encoding block $S_n(x)$, implemented as a Pauli rotation by $x \in [-\pi, +\pi]$, and a trainable block $W_n(\theta_n)$, parametrised by θ_n.

The f_θ circuit acts as as a Fourier-like sum of "frequency" components.

The components are determined by $S_n(x)$ "frequencies" and $W_n(\theta_n)$ coefficients.

Re-uploading of $S_n(x)$ allows to iteratively vary and accumulate rotations $W_n(\theta_n)S_n(x)$.

Pauli rotation gate

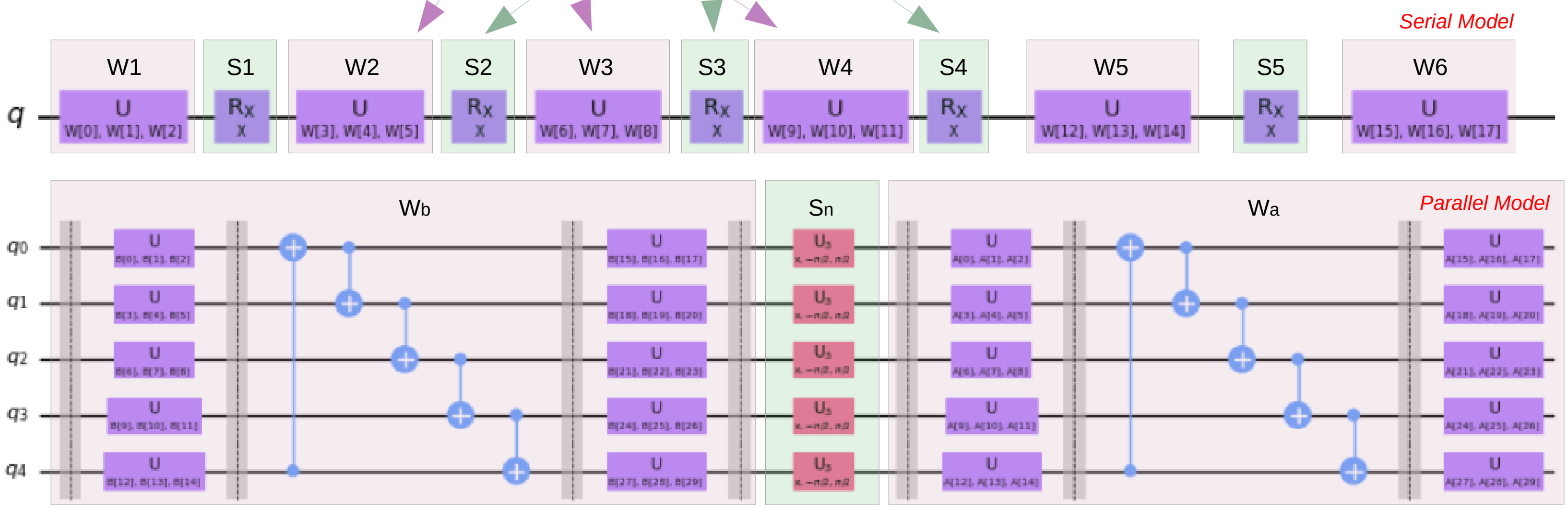
$$S_n(x) = e^{inx}$$

$$f_\theta(x) = \sum_{n \in \Omega} W_n(\theta_n) S_n(x)$$

The circuit can be structured as a *series* of $W_n(\theta_n)\ S_n(x)$ layers, over a single qubit.

Alternatively, $S_n(x)$ blocks can be arranged in *parallel*, with $W_{a/b}(\theta)$ blocks before and after, over multiple qubits.

The circuit parameters can trained by a suitable ML optimiser.



Practice 3
Fourier Transform (Demo)

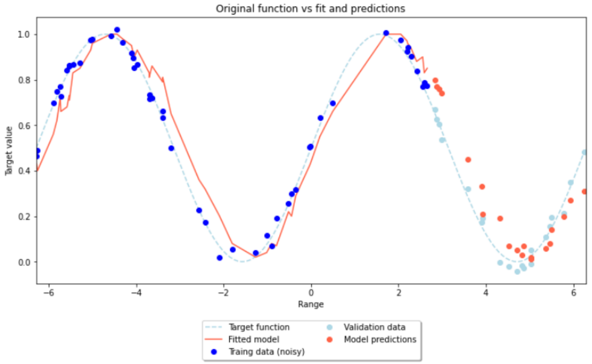
Plan

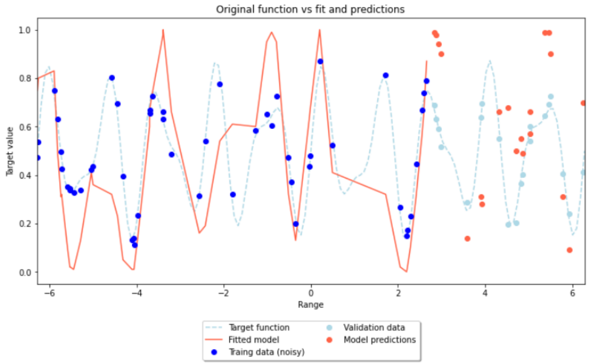
- Navigate own Jupyter Lab
- Walk through Serial Model demo
- Walk through Parallel Model answers (with hidden cells)
- Explain Parallel Model problem
- Explain results on the next page

Experiments F1-8

f(x): sin(x)

f(x): (sin(5x) + 0.5*sin(8x)) / 4 + 0.5





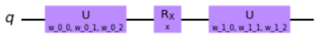
Serial Model

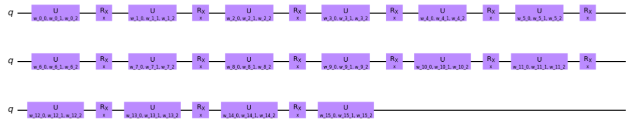
Simple serial model fits the sine function incredibly well!

The fit deteriorates with the number of layers

Note that a simple experiment can show that the serial model directly implements the sine function, so it works so well!

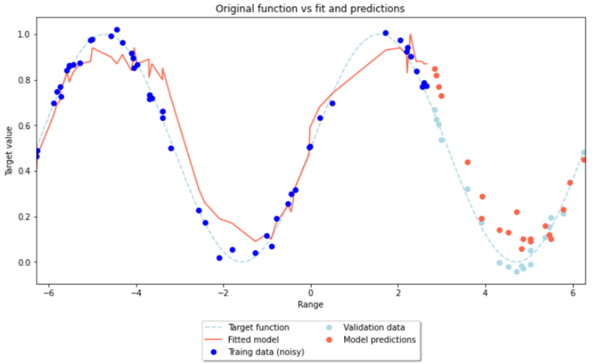
Serial model was not able to cope with any deviation from a simple sine function, e.g. more complex curve or inclusion of a trend, degrades the fit.

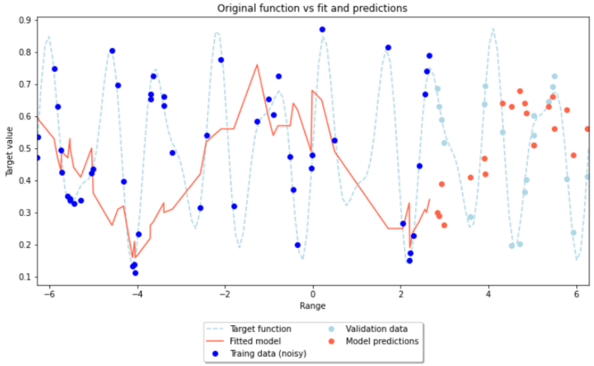




The parallel model does not help (much) with complex functions. Perhaps data re-uploading could be beneficial here?

Parallel Model

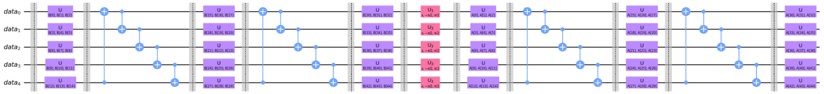


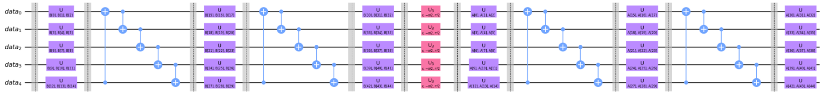


Parallel model greatly improves the performance, however, it is not better than the simplest serial model!

Parallel model degrades with the number of parameters (qubits x layers).

Objective function was highly volatile.



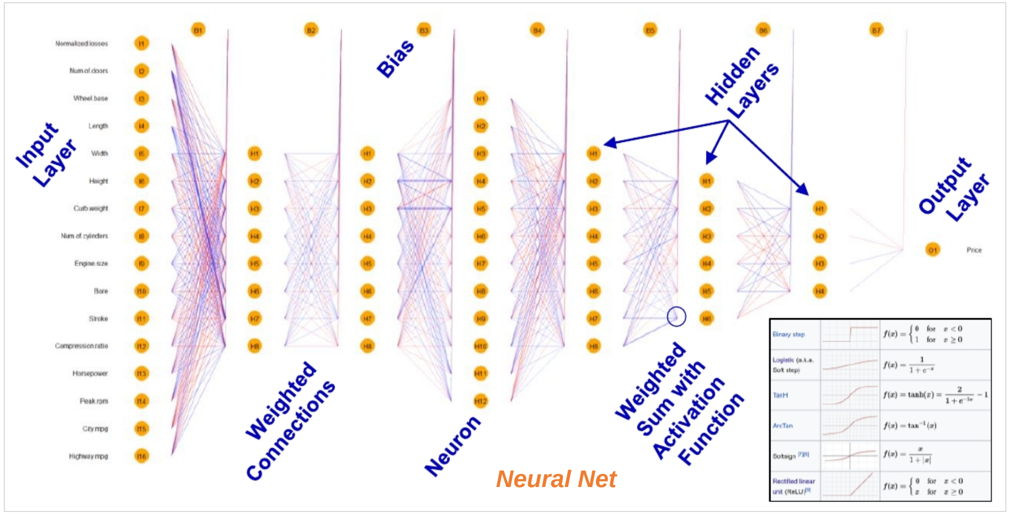


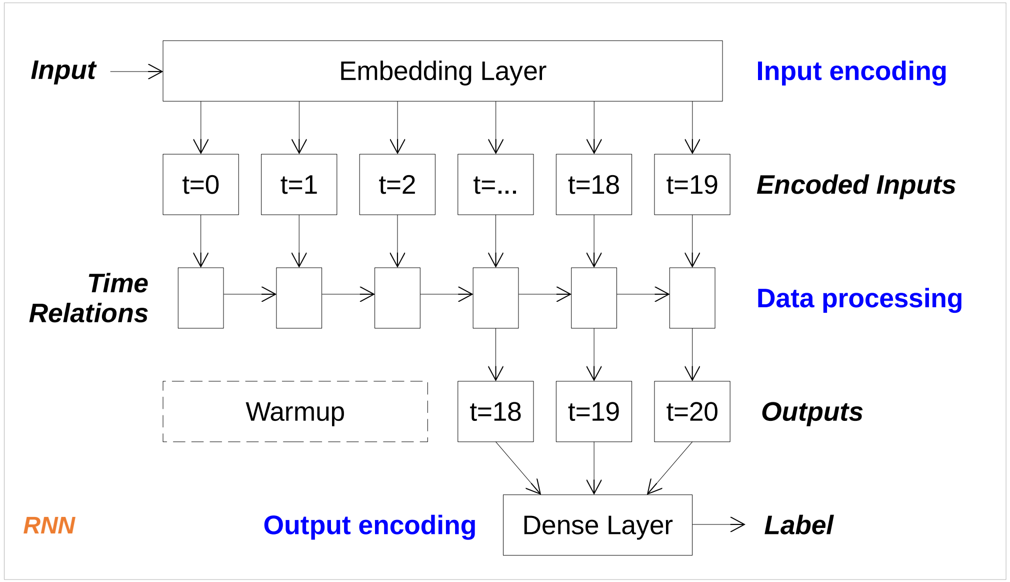
Quantum Fourier transforms Reflection

- Serial quantum Fourier transforms work amazingly well with a single qubit sine curve fitting – not surprising as qubit measurement follows a cosine function
- With increased depth of a serial circuit, the performance decreases
- This is where the parallel quantum Fourier transform steps in and improves the outcome
- However, in both cases the more model parameters, the worse was the outcome (volatility of the objective function)
- Any deviation from a sine function, severely degrades the fit of both approaches
- The hypothesis that the parallel model could improve if we were to adopt the serial model’s data re-uploading proved to be incorrect
- Worth noting that COBYLA and NELDER_MEAD excel in their task, while L_BFGS_B is painfully slow

Neural Nets for Time Series Analysis

- The simplest neural networks, such as Multi-Layer Perceptrons (MLPs), map numeric inputs into numeric or categorical outputs via layers of “neurons”, interconnected by weighed links, and calculating weighted sums with non-linearity
- The weights of neural links are trained within an optimisation process, such as gradient descent, by matching the calculated vs expected outputs
- Some types of deep neural networks can be trained for time series analysis, including: forecasting, classification and clustering, e.g.
  - Recurrent Neural Networks (RNN)
  - Long Short-Term Memory (LSTM) nets
  - Gated Recurrent Units (GRU) nets
- Unlike MLPs, networks such as RNN, LSTM and GRU are able to retain and rely on memory of the past training data
- MLPs and RNNs are similar in their structure to some QTSA quantum solutions (circuits)





Quantum neural networks

Pattern Matching

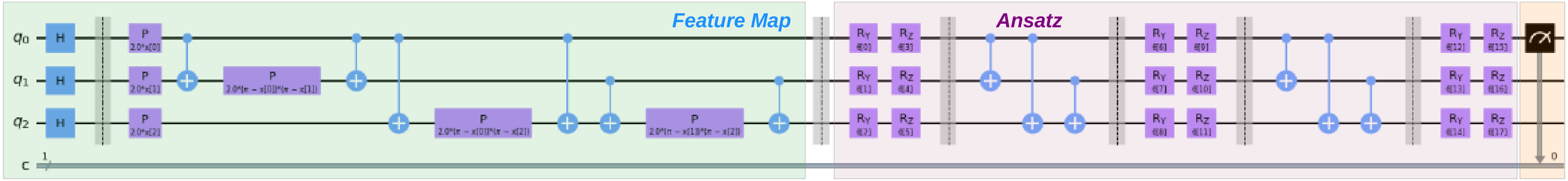
Abbas, Amira, David Sutter, Christa Zoufal, Aurelien Lucchi, Alessio Figalli, and Stefan Woerner. “The Power of Quantum Neural Networks.” Nature Computational Science 1, no. 6 (June 2021): 403–9. https://doi.org/10.1038/s43588-021-00084-1.

Schreiber, Amelie. “Quantum Neural Networks for FinTech.” Medium, May 8, 2020. https://towardsdatascience.com/quantum-neural-networks-for-fintech-dddc6ac68dbf.

- A typical QNN consists of two main components, i.e. a feature map and an ansatz (also called variational model)
- The feature encodes the input data and prepares the quantum system state, using as many features as there are qubits
- The ansatz consists of several layers and, similarly to a classical NN, is responsible for inter-linking the layers - this is accomplished by trainable Pauli rotation gates and entanglement blocks
- Finally, the qubit states are measured and interpreted as QNN output

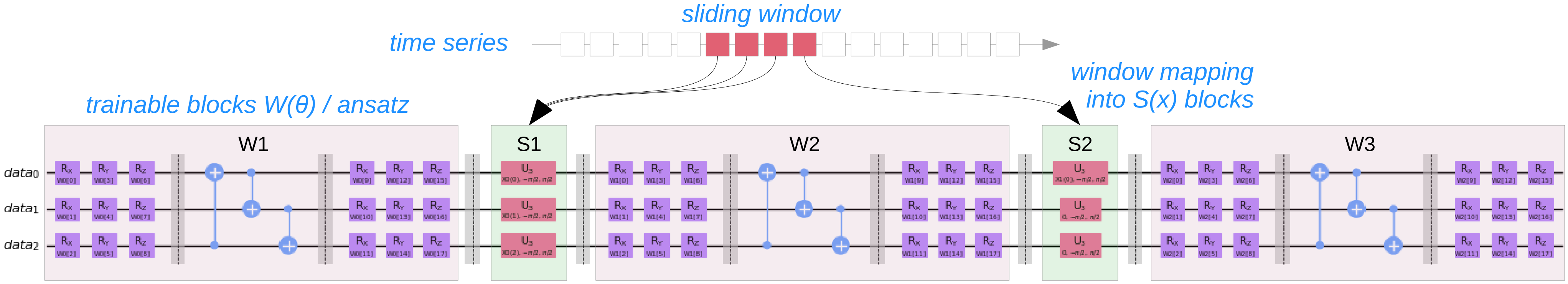
- In contrast to function / data fitting, QNNs are able to perform pattern matching, i.e. work with a sequence of values themselves rather than with the mapping between an index and values
- In the following experiments, we will adopt a sliding window approach to structuring the time series
- However, the standard QNN model does not lean itself to time series analysis, i.e.
  - You are limited to the TS window of size equal to the number of qubits

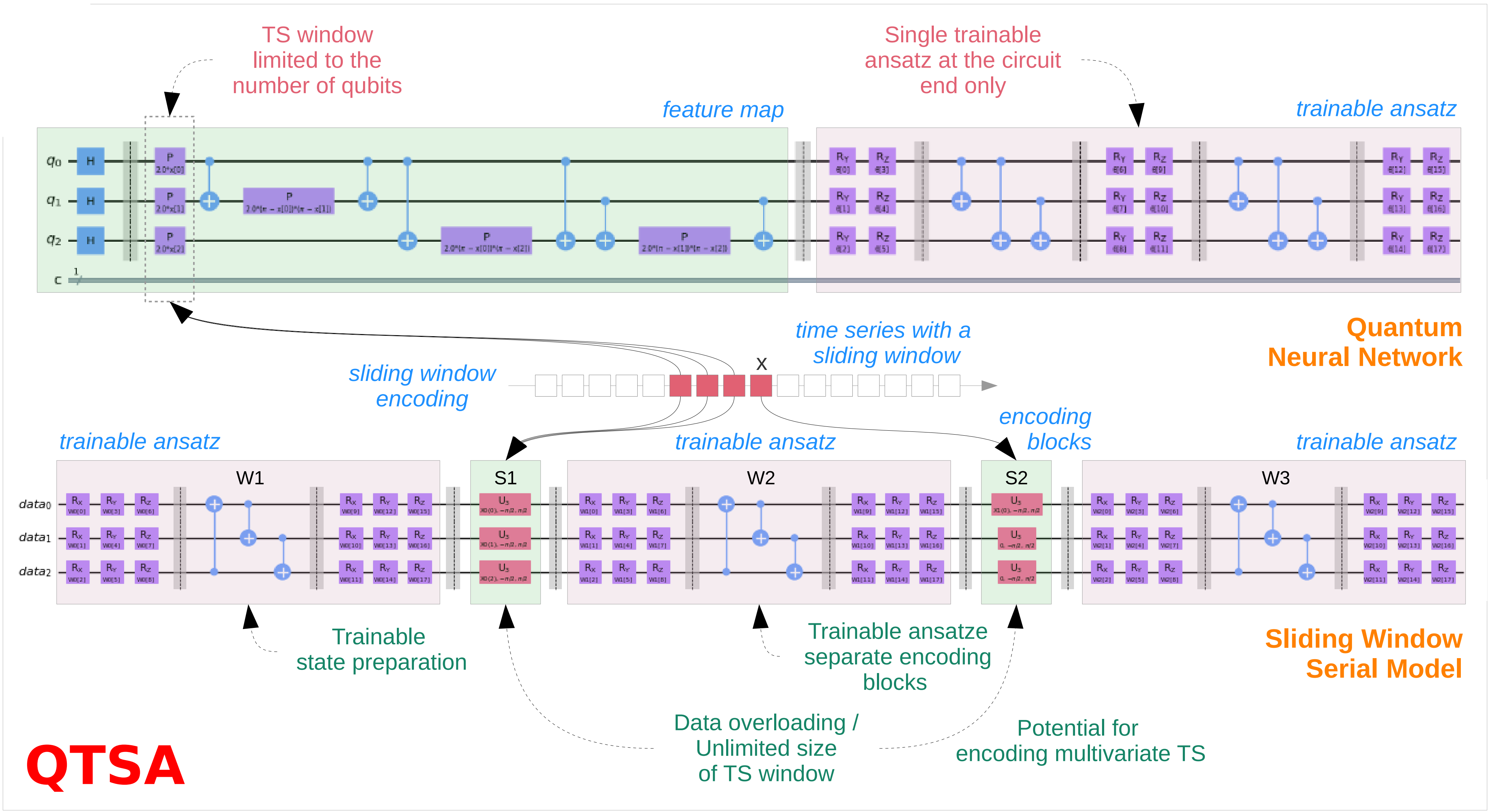
In Qiskit VQR Model



QNN inspired QTSA
Sliding windows / Serial model

- Experiments show that typical QNN (VQR) do not perform well with time series data
- The solution is to extend the Fourier quantum model into the multi-qubit QNN
- This required creation of a custom quantum circuit, which consists of encoding blocks $S_n(x)$ and trainable ansatz blocks $W_n(\theta_n)$
- The Fourier parallel model simply replicated the Sn(x) blocks, which limited the TS window size to the number of qubits, and which was tested to perform quite poorly
- An alternative was to adapt the Fourier serial model and distribute the TS window data across the encoding blocks $S_n(x_k)$, where each block would hold as many data points as there are qubits (k)
- Should the last block $Sn(x_k)$ be only partially filled with TS data, then the identity gates are used to make the complete block
- The circuit is then trained by optimising the parameters of trainable blocks $W_n(\theta_n)$

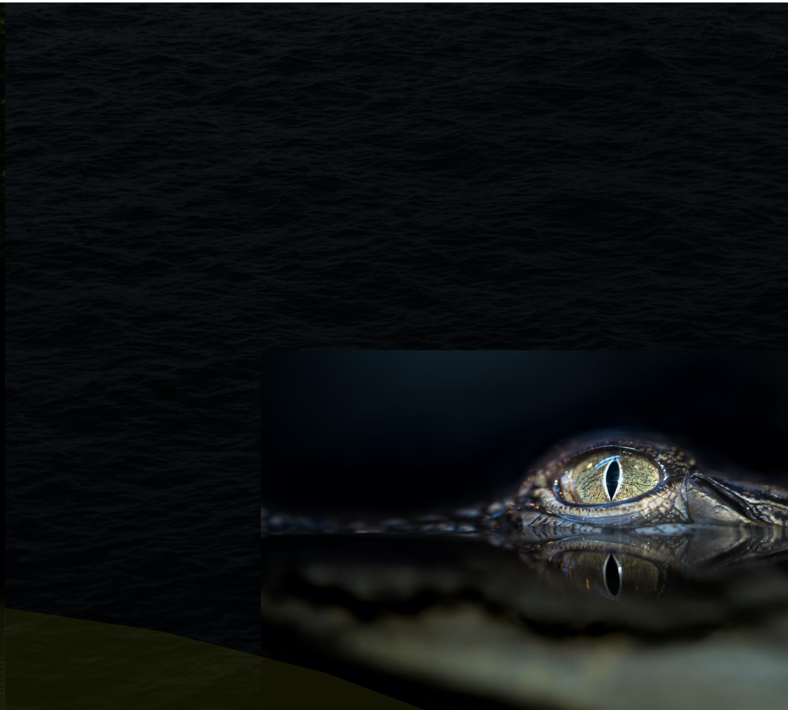



TS window
limited to the
number of qubits
Single trainable
ansatz at the circuit
end only
feature map
trainable ansatz
Quantum
Neural Network
time series with a
sliding window
X
sliding window
encoding
encoding
blocks
trainable ansatz
trainable ansatz
trainable ansatz
W1
S1
W2
S2
W3
Trainable
state preparation
Trainable ansatze
separate encoding
blocks
Sliding Window
Serial Model
Data overloading /
Unlimited size
of TS window
Potential for
encoding multivariate TS
QTSA

Practice 4
QNN Inspired QTSA (Demo)

Plan

- Explain there is standard QNN demo
- Walk through Sliding Window answers (with hidden cells)
- Explain Sliding Window problem
- Explain results on the next page



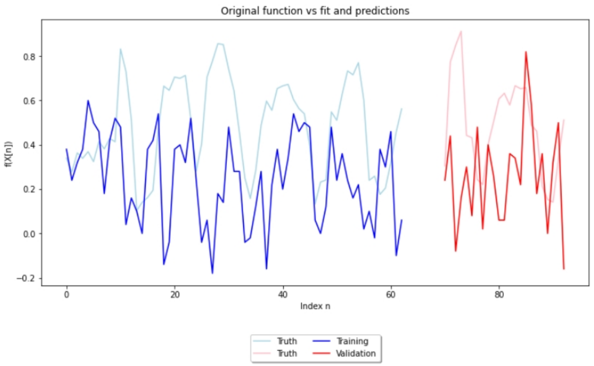
Experiments N1-N8

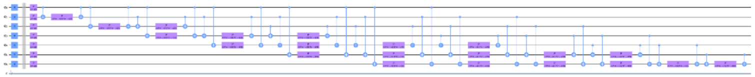
f(x): (sin(5x) + 0.5*sin(8x)) / 4 + 0.5

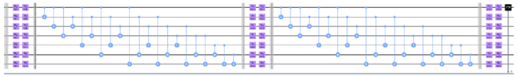
QNN model worked well only for simple data sets and shorter TS windows.

With more complex data and longer TS windows, its performance significantly degraded.

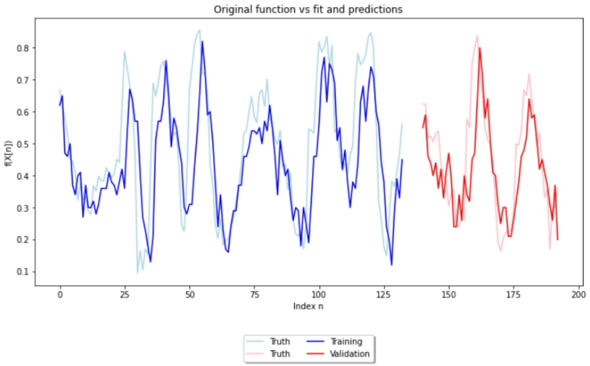
Sliding Window QNN

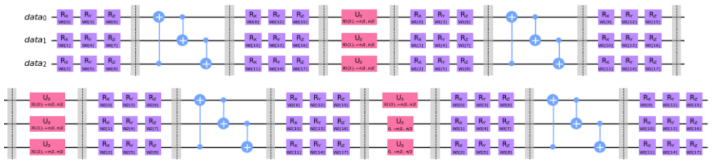






Sliding Window Serial Model





Sliding window serial model demonstrated good prediction of both seasonality and the signal amplitude.

The model does not re-upload data, however, it overloads TS data points.

This model can handle TS windows longer than the number of qubits available!

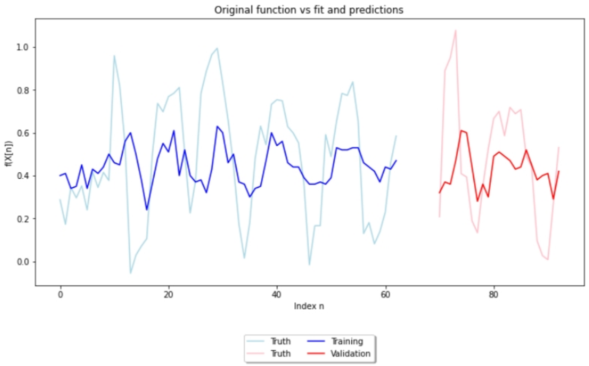
The model prediction significantly improved, well above the classical MLP!

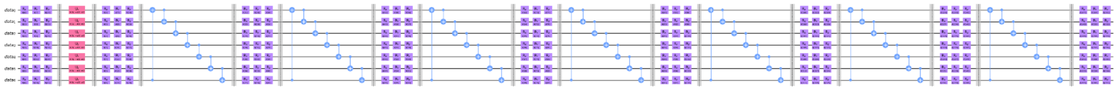
Sliding window parallel model displayed pretty good prediction of seasonality.

However, its failed predicting the data amplitude.

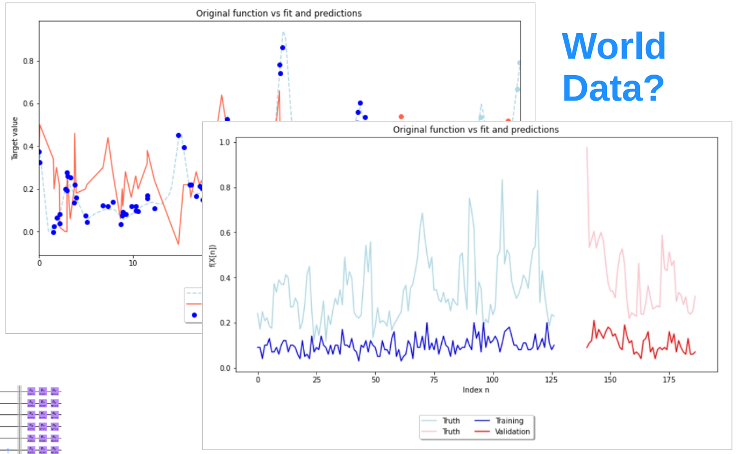
Parallel models tend to generate very large circuits, which may adversely affect the results.

Sliding Window Parallel Model





World Data?



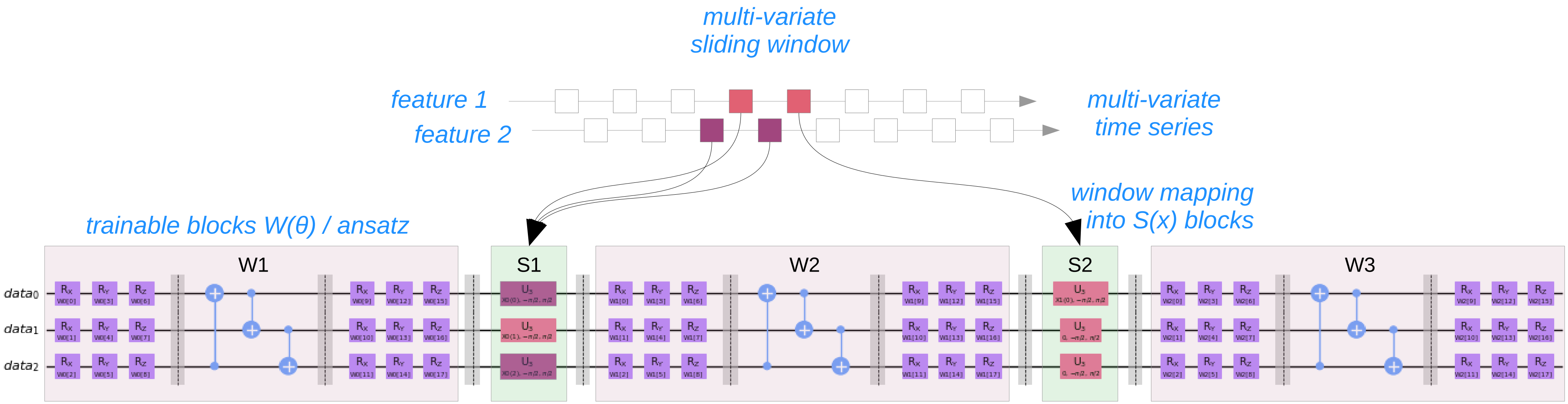
Real-world data, such as records of beer sales in USA, also featured in our research.

The preliminary experiments indicate that more work is required to make the proposed model practical.

The proposed methods need new circuit measurement strategies and inclusion of non-linearities.

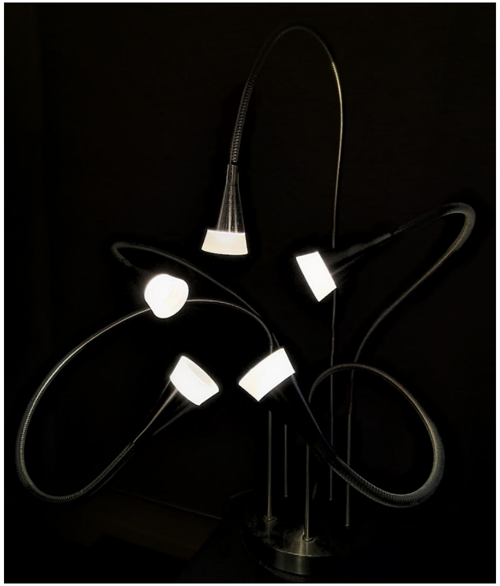
Opportunities for Multivariate TS Analysis

- The serial model with a sliding window accommodates multi-variate time series
- Trainable ansatz blocks $W_n(\theta_n)$ are generated in exactly the same way as previously
- However the encoding blocks $S_n(x)$ interweave the feature values taken from the multi-variate sliding window



Quantum neural networks Reflection

- QNNs are a promising approach to QTSA and forecasting.
- Unlike other QTS methods, QNNs are capable not only of data fitting but also pattern matching and prediction.
- However, the standard QNN model consisting of a feature map and a trainable ansatz, demonstrate poor performance when trained with more complex data.
- The proposed model, extends the single-qubit quantum Fourier serial model to work with multiple qubits and with complex data.
- The model relies on the TS data continuity to reduce the need for re-uploading its input data, and instead overloads the qubits with blocks encoding of the entire TS window of data points.
- The model is able to encode more data than the number of its qubits, the preliminary experiments demonstrate its performance approaching that of the classical MLP, and they can accommodate multivariate time series.



Other researchers in QNNs proposed quantum models of RNNs and LSTM!

Bausch, Johannes. “Recurrent Quantum Neural Networks.” *Advances in Neural Information Processing Systems* 33 (2020): 1368–79.

Chen, Samuel Yen-Chi, Shinjae Yoo, and Yao-Lung L. Fang. “Quantum Long Short-Term Memory.” In ICASSP 2022-2022 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8622–26. IEEE, 2022.

Things we have not covered...

Published (preliminary) research

- Quantum stochastic / Bayesian TS models (e.g. random walk)
- Quantum anomaly detection in time series
- Quantum RNN and LSTM
- qGANs and quantum variational auto-encoders for QTSA
- Quantum natural language analysis (text is a sort of TS)
- Applications, e.g. in finance, medicine, signal and image processing

Other opportunities

- Non-stationary time series
- Multi-variate time series analysis
- Volatility detection in time series
- Chaotic behaviour over time
- Quantum/classical scenario analysis
- Creation of pure quantum QTSA
- And more…

Quantum time series analysis is still in a barren plateau of QML research!

Why in this session haven't we seen quantum advantage of QTSA models?

As many other QML models!

- This session in two parts focused on the fundamentals of quantum time series encoding, processing and measurement.
- As you recall quantum applications, which successfully compete against classical approaches, take advantage of three quantum phenomena, i.e. state superposition, entanglement and state collapse on measurement.
- All QTSA models presented here have taken advantage of these quantum phenomena and yet did not perform better than the classical systems!
- At the same time note that the presented quantum solutions:
  - Were created by brute force
  - Had no theory of their data
  - Did not prepare their data
  - Did not plan their measurement
  - Had ad hoc processing
  - We made no attempt to detect / eradicate any training issues
- Quantum advantage of QTSA models is still part of active research.

To gain quantum advantage in QTSA will be the focus of our future Part 3

Bird-view of Quantum Time Series Analysis
Summary, reflections and questions

TS processing requires data storage

Variational quantum regression is too simplistic

QNNs with data re-uploading and overloading are key

Quantum systems have no memory

Quantum Fourier transforms are promising

Variational quantum models effectively simulate memory

QC creates opportunities for TSA

Quantum neural nets suggest the solution to QTSA

QTSA Workshop Session Tasks

In your own time!

1) Easy: Study serial quantum Fourier transform TS fit.
   Test it with various data sets, factors and optimisers.
   Analyse, compare and find the best combination.
2) Medium: Implement a parallel quantum Fourier transform TS fit.
   Test it with various data sets, factors and optimisers.
   Analyse, compare and find the best combination.
3) Hard: Implement a serial sliding window QNN TS forecaster.
   Test it with various data sets, factors and optimisers.
   Analyse, compare and find the best combination.
4) Challenge: Modify the SSW QNN for multi-variate TS data.
   Test it with various data sets, factors and optimisers.
   Analyse, compare and find the best combination.

End of Part 2

All images are either the author's own, open source, or
from the internet's source of freely usable images: https://unsplash.com/