Universidad de Granada

decsai.ugr.es

Time Series forecasting with Quantum Neural Networks

Manuel Pegalajar Cuéllar, IWANN'23

DECSAI

Departamento de Ciencias de la Computación e Inteligencia Artificial

DECSAI

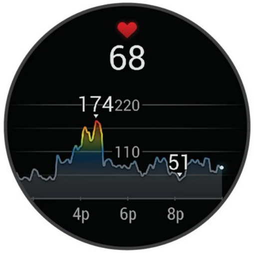
DECSAI

Definition of Time Series

A Time Series ***X(t)= {x(1), x(2), x(3), …, x(t)}*** is a sequence of observations of a given phenomenon, sampled **periodically** and **indexed** in time.

Examples of Time Series :

- *Prices of stock markets*
- *Electricity usage*
- *History of weather information*
- *Sensor activity monitoring*
- *Heart rate*
- *Etc.*



The forecasting problem

Let $X(t) = \{x(1), x(2), \ldots, x(t)\}$ be a sequence of historical data of a given phenomenon, sampled periodically and indexed in time. The ***forecasting*** problem attempts to predict $x(t+1)$ as :

$$x(t+1) = f(x(t), x(t-1), \ldots, x(t-T), w) + \varepsilon(t+1)$$

Where :

- $f()$ is a model hypothesis
- T is a history time horizon
- $w = (w_1, \ldots, w_n)$ are the model parameters
- $\varepsilon(t+1)$ is the approximation error to $x(t+1)$ using $f()$ (unknown)

Example

To predict the energy consumption of a building for the next monday, having into consideration the daily energy consumption historical data of the past week (**T=7**).

ANNs for Time Series forecasting

Feedforward ANNs : *Time component removed*

The time series values ***{x(t-1), x(t-2), …, x(t-T)}*** are organized as tabular data and used as network inputs. The network output is (supposed to be) ***x(t)***.

Recurrent ANNs : *Time dependency considered*

The time series values ***{x(t-T), x(t-T+1), …, x(t-1)}*** are fed sequentially to the network (one value at a time) are used as network inputs. The network output is (supposed to be) the next time series values ***{x(t-T+1), x(t-T+2), …, x(t)}.***

The objective of our research

Quantum Computing for Time Series forecasting

- How Quantum Computing (QC) can be used to solve Time Series forecasting problems ?
- Is there any practical advantage of QC to solve this problem ?

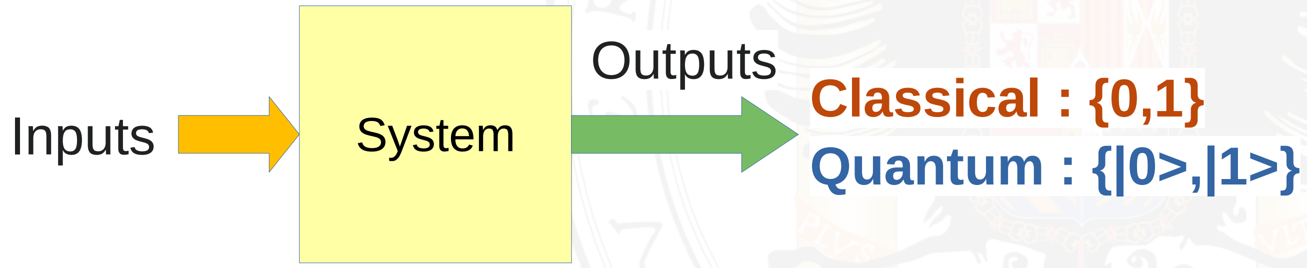
Classical vs Quantum computing from an external observer

Classical computing :

- Binary outputs (bits, {0,1})
- One bit : Operations over Z_2 (the field with two elements)
- n bits : Operations over Z_2^n (cartesian product)

Quantum Computing

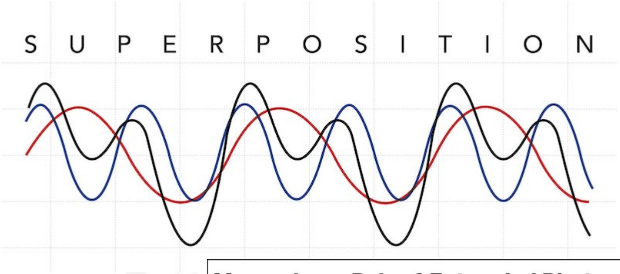
- Binary outputs (qubits, {|0>,|1>})
- One qubit : Operations over C^2 (the complex plane)
- n qubits : Operations over C^{2^n} (tensor product)

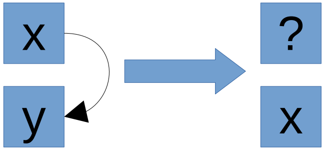


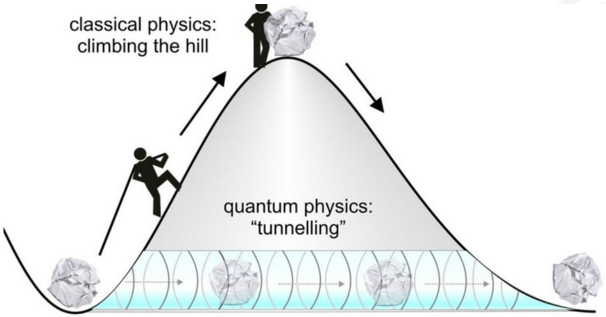
Classical vs Quantum computing (internals)

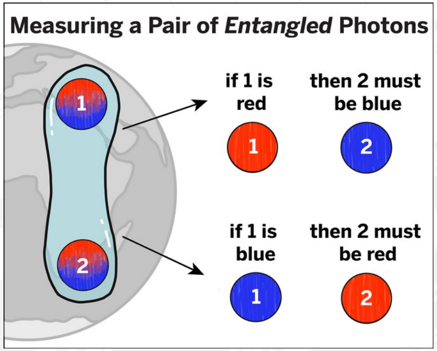
New in Quantum Computing :

- Superposition,
- Quantum parallelism,
- Entanglement,
- Tunnelling,
- Teleportation, ...









What I personally miss from Classical Computing :

- All operations **must be reversible** in QC.
- No copy (i.e., x=|1> is allowed ; x=y is **NOT** allowed) : Teleportation instead.
- No loops nor jumps backwards, ...

One qubit (*maths*)

- **Qubits |0>, |1>** : The orthonormal basis of a complex vector subspace in C^2 (column vectors). The **computational basis**.

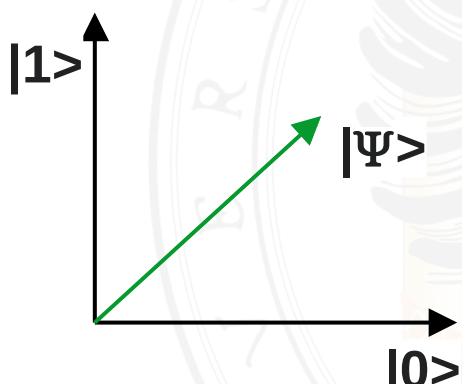
$$|0> = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; |1> = \begin{pmatrix} 0 \\ 1 \end{pmatrix}; |0>, |1> \in C^2$$

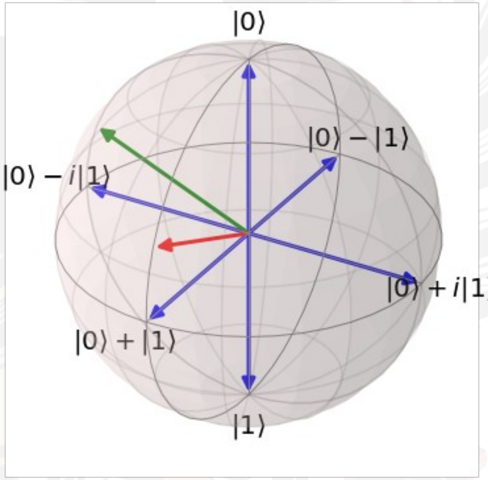
- **Arbitrary qubit |Ψ>** : Linear complex combination of basis vectors : Infinite possible values before measurement.

$$|\psi> = \alpha_0 |0> + \alpha_1 |1>, \alpha_0, \alpha_1 \in C$$

Subject to :

$$\sum_i |\alpha_i|^2 = 1$$





n qubits *(maths)*

- The orthonormal basis of a system with ***n*** qubits is calculated as the tensor product of the computational basis of one qubit.
- The size of the computational space doubles its size with each new qubit, up to C^{2^n}.

Example for 2 qubits :

$$|0>\otimes|0>=\begin{pmatrix}1\\0\end{pmatrix}\otimes\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}1\begin{pmatrix}1\\0\end{pmatrix}\\0\begin{pmatrix}1\\0\end{pmatrix}\end{pmatrix}=\begin{pmatrix}1\\0\\0\\0\end{pmatrix}=|00>$$

$$|1>\otimes|0>=\begin{pmatrix}0\\1\end{pmatrix}\otimes\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}0\begin{pmatrix}1\\0\end{pmatrix}\\1\begin{pmatrix}1\\0\end{pmatrix}\end{pmatrix}=\begin{pmatrix}0\\0\\1\\0\end{pmatrix}=|10>$$

$$|0>\otimes|1>=\begin{pmatrix}1\\0\end{pmatrix}\otimes\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}1\begin{pmatrix}0\\1\end{pmatrix}\\0\begin{pmatrix}0\\1\end{pmatrix}\end{pmatrix}=\begin{pmatrix}0\\1\\0\\0\end{pmatrix}=|01>$$

$$|1>\otimes|1>=\begin{pmatrix}0\\1\end{pmatrix}\otimes\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}0\begin{pmatrix}0\\1\end{pmatrix}\\1\begin{pmatrix}0\\1\end{pmatrix}\end{pmatrix}=\begin{pmatrix}0\\0\\0\\1\end{pmatrix}=|11>$$

Quantum operations

Classical gates:

Implemented with the +,· operators of the underlying field Z_2.
They create a new result, input parameters are not modified.

$$\mathrm{AND}(x,y)=x\cdot y\,; \quad \mathrm{OR}(x,y)=x+y+x\cdot y$$

Quantum gates:

Modelled as unitary matrices that multiply the system (quantum) state. They **evolve** the whole state to a new one.

$$CNOT=\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Initial quantum state : $|10>$

$$CNOT|10>\rightarrow|11>$$

Quantum operations

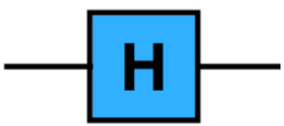
Some quantum gates (all reversible):

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

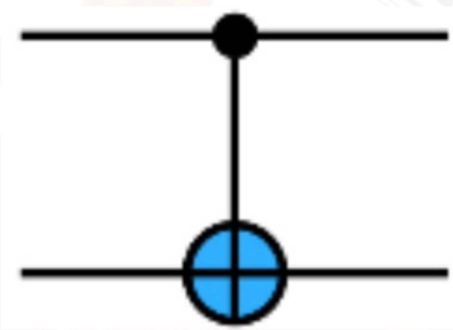
$$R_\phi^Z = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}$$

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$H = \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{pmatrix}$$



$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$



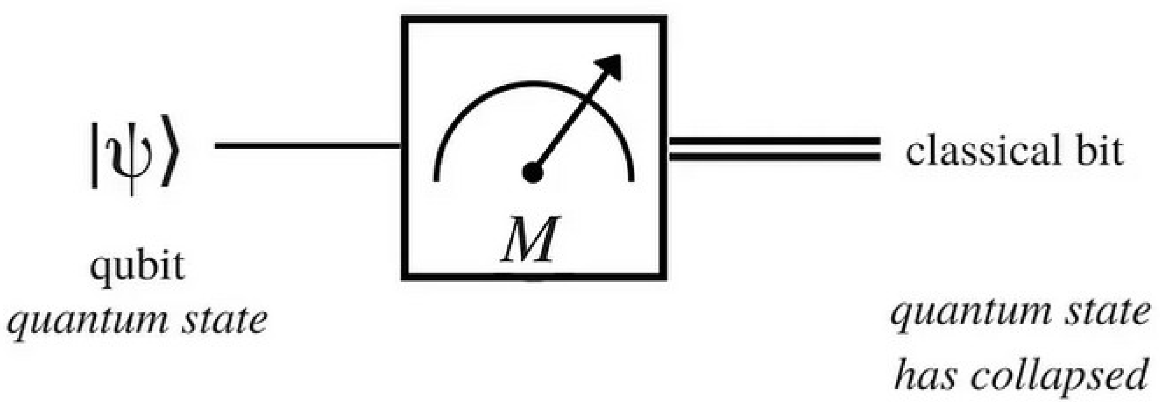
Quantum operations

The measurement operator (not reversible):

If an arbitrary qubit is modelled as : $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle, \alpha_0, \alpha_1 \in C$

Then, measurement makes the state to collapse and returns :

- Value |0> with probability $|\alpha_0|^2$
- Value |1> with probability $|\alpha_1|^2$



Quantum algorithms and circuits

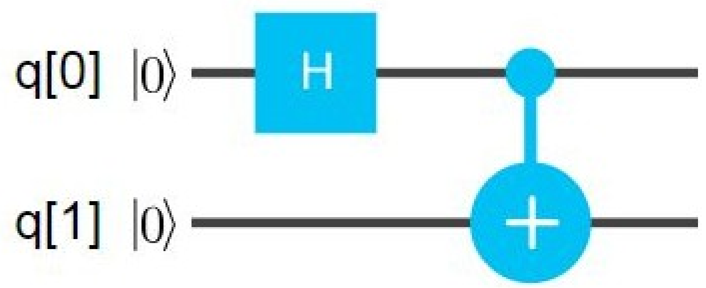
A quantum algorithm is a sequence of quantum gates (and possibly measurement operators). It is implemented in a quantum circuit whose graphical representation is :

- An horizontal line for each qubit
- Gates have their own graphical representation.
- Gates involving more than one qubit link sources/controls and targets with vertical lines.

Example :

Quantum circuit to build the quantum algorithm :

$$CNOT\left(\left(H|q_0>\right)\otimes|q_1>\right)$$

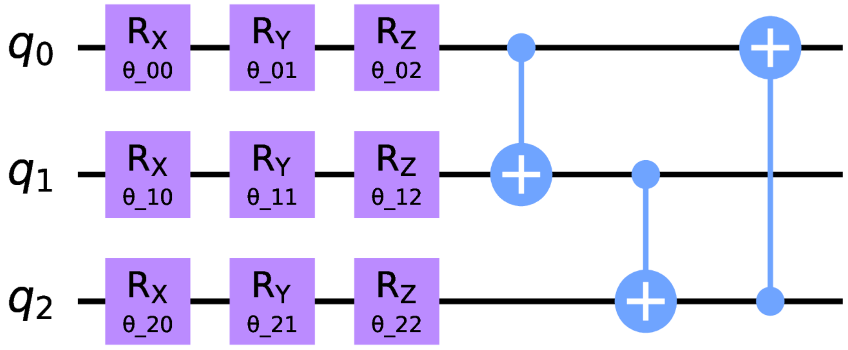


Variational Quantum Circuits (VQCs)

Circuits with parameterized gates, such as Rx(Θ), Ry(Θ), Rz(Θ) with parameter Θ.

The behaviour of a quantum circuit might vary depending on the gates ‘ parameters.

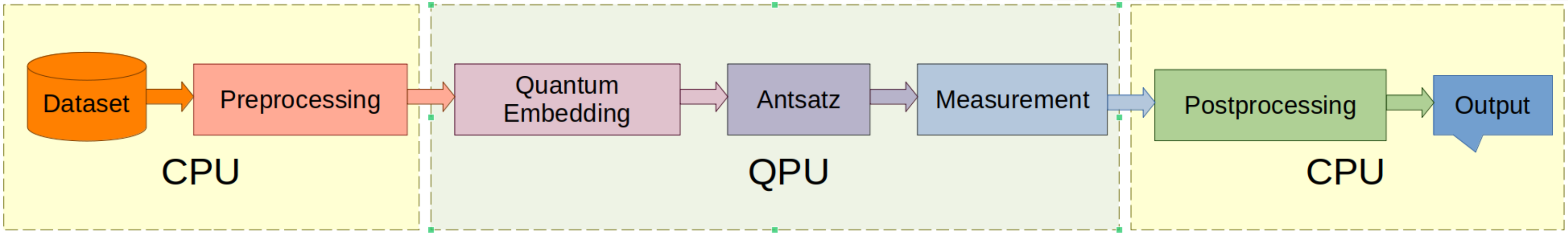
Fundamental block of **Quantum Machine Learning**.



Quantum Machine Learning (QML)

It attempts to move the ideas of classical Machine Learning to the QC paradigm.

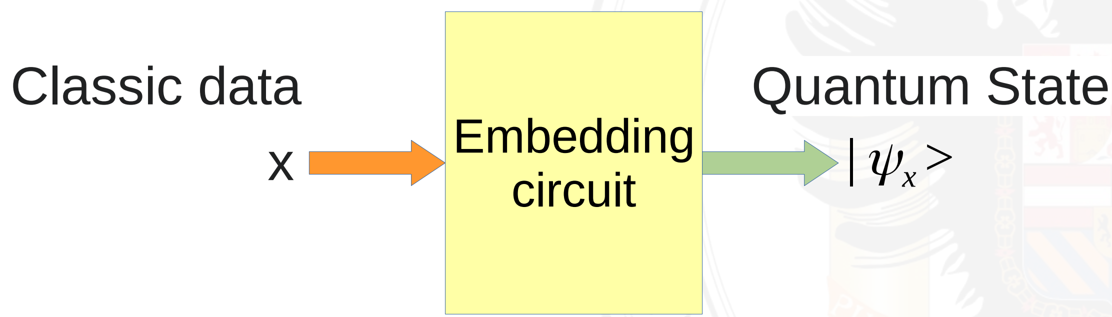
- ***Supervised Learning***
- ***Unsupervised Learning***
- ***Reinforcement Learning***



Quantum Embedding

It is used to encode a classic data into a quantum state

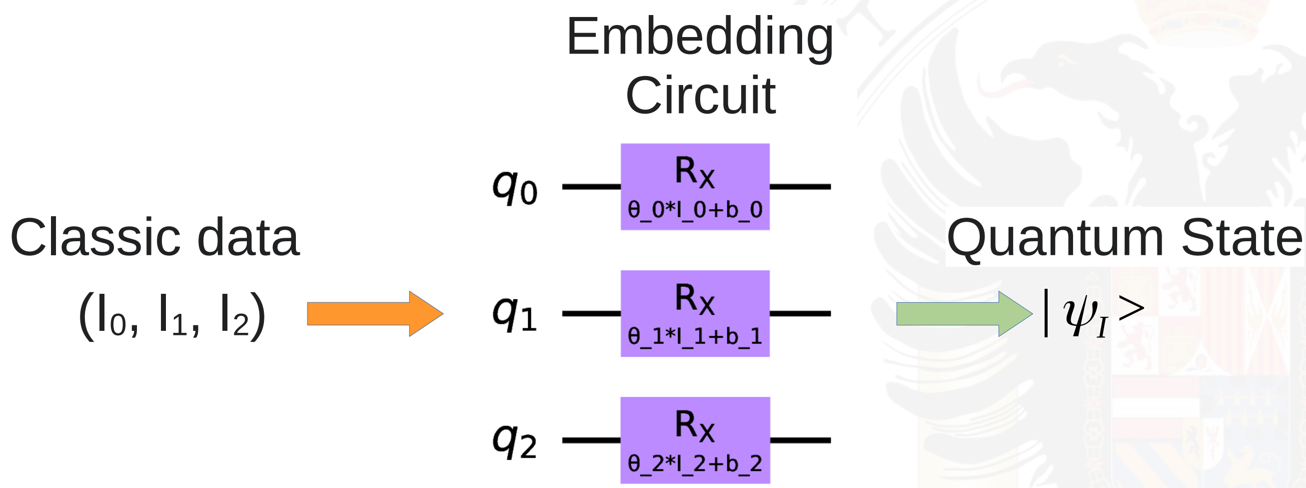
- ***Basis encoding***
- ***Q-Sample encoding***
- ***Amplitude encoding***
- ***Angle encoding***
- ***Tensor product encoding***
- ***...***



Quantum Tensor Product Embedding

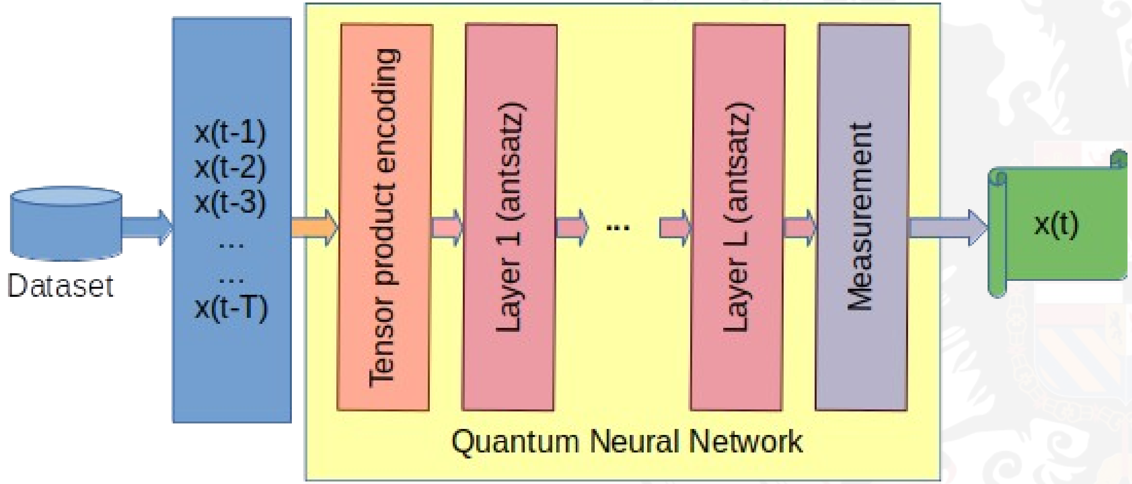
Parameterized X-Rotation gates are used to encode n-dimensional data into n qubits.

In our approach :



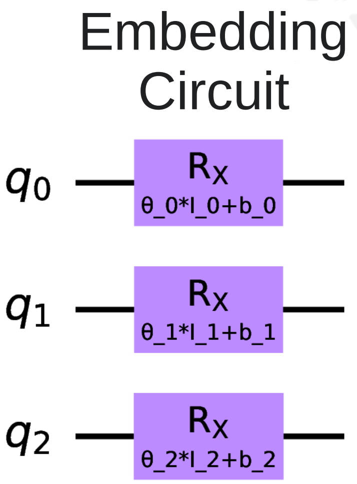
General model

- Time Series data are transformed to tabular data with T past values as inputs to predict the next value.
- Inputs **{x(t-T+1), x(t-T+2), …, x(t-1)}** are fed to the Quantum Circuit using Tensor Product Encoding.
- A VQC is the analog to a network layer.
- Final measurement using the expectation of the quantum state with the observable Z is used to obtain the desired value **x(t)**

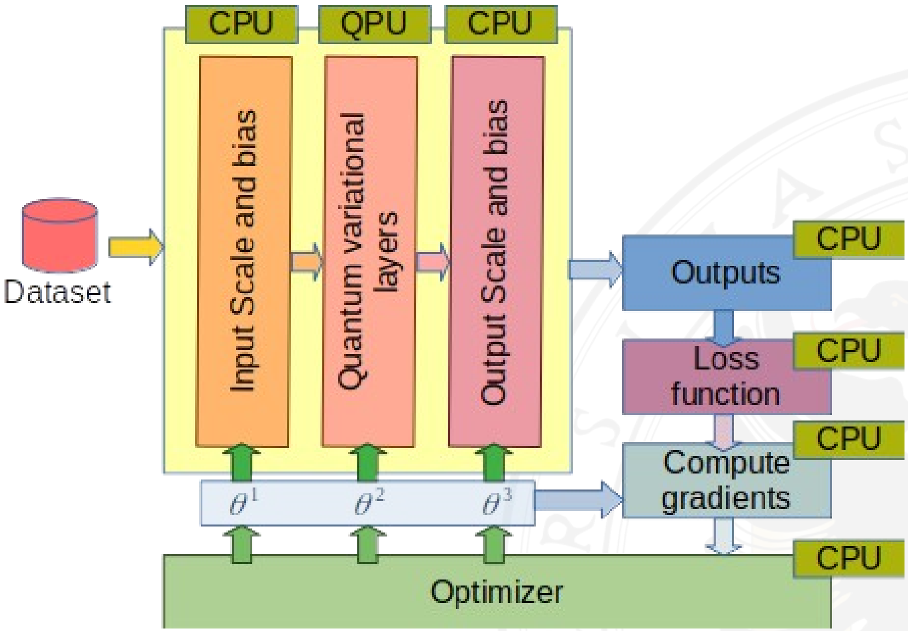


General model tuning

- Both input and output values are biased and scaled to make training easier.
- Consequence : We have both classical and quantum parameters to be optimized.

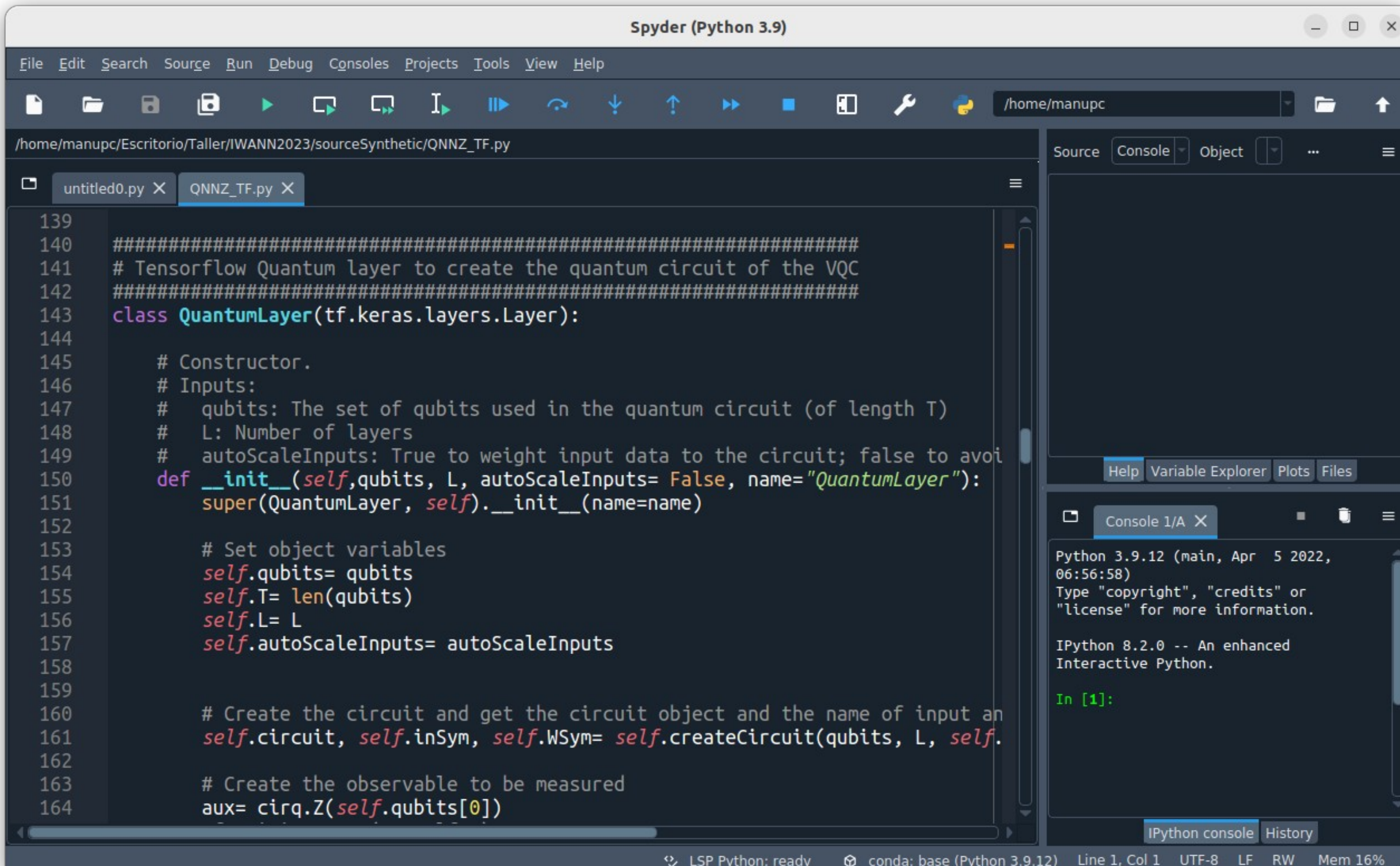


Training scheme



Source code available online

https://github.com/manupc/qnn_tsp

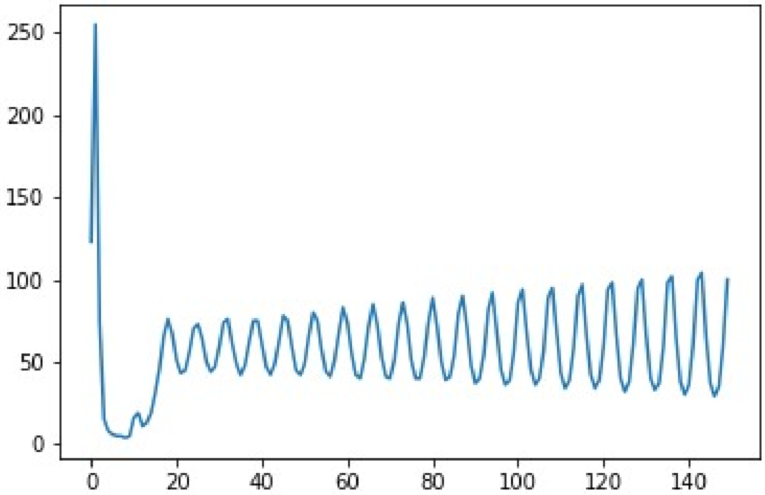


DECSAI

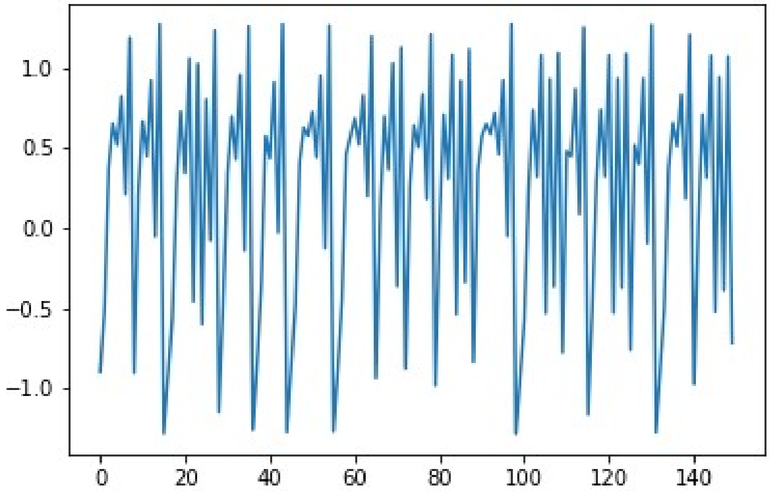
Datasets

- Goal : To test the performance of VQCs for Time Series forecasting.
- Two traditional well-known time series benchmarks are used.
- Time Series length : 150 values (75 % training, 25 % test)
- 4-Fold Cross-Validation
- Preprocessing : Change of scale to [-1, 1]

Laser data



Henon data



Comparison

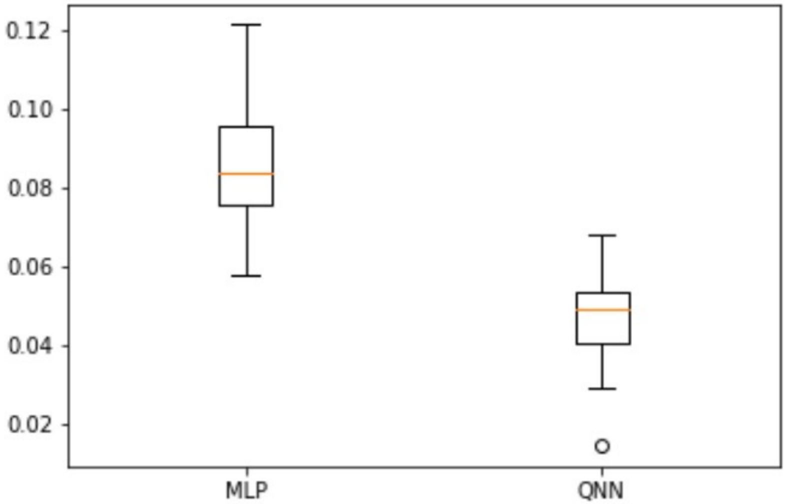
- VQCs are feedforward models
- A Multilayer Perceptron (MLP) was used

Results (after 30 executions)

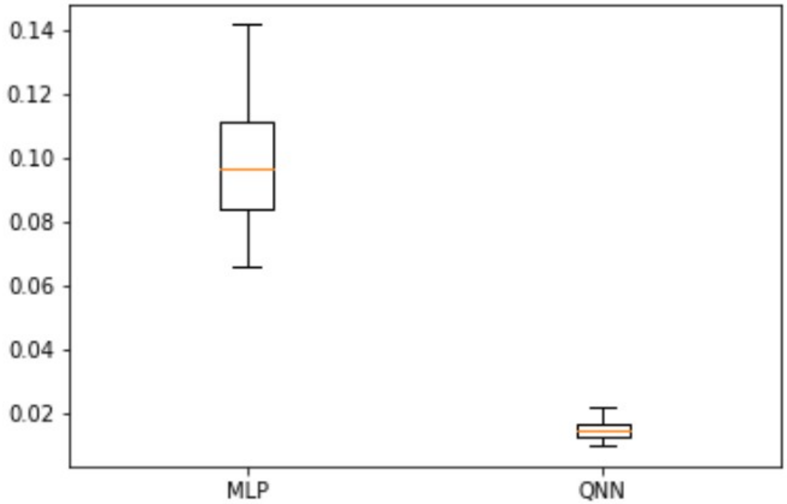
Metric	Laser		Henon	
	MLP	QNN	MLP	QNN
Avg. Tr. MSE	0.0380	0.0168	0.0761	0.0044
Avg. Ts. MSE	0.0478	0.0199	0.0734	0.0063
Avg. Val. MSE	0.0863	0.0476 (+)	0.0977	0.0145 (+)
Min. Val. MSE	0.05774	0.01461	0.0655	0.0096
Max. Val. MSE	0.1215	0.0683	0.1416	0.02145
Avg. Time	10.36	30.92	10.44	30.89

Results – Agv. Error in test - (boxplots after 30 executions)

Laser data



Henon data



UNIVERSIDAD DE GRANADA

DECSAI

Lessons learned

- QML models can be used to solve Time Series forecasting problems.
- Finding a QNN topology in QC is harder than in classical ML
- It is possible to overcome local optima with respect to traditional methods (MLP).
- The number of parameters (model complexity) can be significantly smaller than classical models.
- The computational complexity (experimental time) is significantly larger in QC simulation software.

Future works

- Simulation of Recurrent Neural Networks.
- To find a more compact data representation able to take advantage of dense encoding.
- Application to other time series problems (classification).

Universidad de Granada

decsai.ugr.es

Time Series forecasting with Quantum Neural Networks

Manuel Pegalajar Cuéllar, IWANN'23

DECSAI

Departamento de Ciencias de la Computación e Inteligencia Artificial