Feed Forward Photonic Circuits for Quantum Machine Learning

Stephen Collins

July 2023

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`. I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at `https://libguides.tcd.ie/plagiarism/ready-steady-write`.

1 Abstract

Using variational quantum algorithms for machine learning is viewed as one of the most practical near-term applications of quantum computing. In this report, I explore the potential for using linear optics, supplemented with feed-forward circuits, for quantum machine learning. Quantum circuits can be used to approximate a Fourier series, and analysis of this sheds some light on how feed-forward circuits can provide more accuracy in learning to model certain target functions. Interestingly, simulations and analysis have shown feed-forward circuits to be more expressive than other comparable regimes for certain tasks. This work leaves open the possibility of running QML models using feed-forward regimes on a QPU at Quandela, as well as exploring deeper circuits and more complex problems.

2 Introduction

Quantum Machine Learning (QML) is a fast evolving field within Quantum Computing (QC) and has been shown to have practical applications in the current Noisy Intermediate Scale Quantum (NISQ) era [1, 2, 3]. In this report, I will first introduce how linear optics can be used to create a quantum device, and explore the advantages of using photons as quantum information carriers compared to other implementations. I then investigate how Measurement-Based Quantum Computing (MBQC) can lead to Feed Forward (FF) architectures which can be used for ML routines, specifically in modelling Fourier series. I also outline how we can measure QML models and the current challenges in generalising learning tasks, and in comparing different schemes. In the Methods section, I take a deeper look at how Fourier series can be modelled using a photonic circuit, and how a feed forward implementation can create a more powerful model. Finally, I present some schemes, using Quandela's software package Perceval, which show promising results when using FF circuits to model problems.

2.1 Linear Optics

Linear optical quantum computing (LOQC) uses photons as quantum information carriers, by encoding a photon into a mode of a quantum device. Encoding photons into spatial modes is known as path encoding. By using one photon and two modes, we can define a qubit in path encoding by assigning each mode as a basis state for the photon [4] (i.e. $\{|0,1\rangle, |1,0\rangle\}$ corresponds to the logical computational basis $\{|0\rangle_L, |1\rangle_L\}$. Computations can then be performed as unitary operations via optical components such as waveplates, beam splitters and phase shifters. Alternatively, we can use a photon's internal spin polarisation as another degree of freedom to encode information. This is known as polarisation encoding [5]. For the rest of this report, we will use path encoding for the purpose of quantum information processing (QIP) using photonics.

Using path encoding in photonics provides a key difference compared to other QIP implementations (such as superconducting qubits or trapped ion qubits). In a typical two-state quantum system, each output wire can be characterised as a complex Hilbert subspace of two dimensions (e.g. $\mathbf{C}_2$). The total space is then a tensor product of each subspace (e.g. $\mathbf{C}_2 \otimes \ldots \otimes \mathbf{C}_2$). In a photonic architecture, this corresponds to using each pair of modes as an output wire, restricted to the computational basis above.

However, if we allow the photons to reside in any mode, this increases the dimension of the space (each output wire is now a mode containing up to n photons). The total space is then a Fock space. Consider the case where our system has n photons and $2n$ modes. Using the qubit characterisation, this system has a dimension of 2^n. However, in Fock space, our system dimension increases binomially with n, and is equal to $\binom{n+m-1}{n}$ Fig. 1 shows how each space grows with n.

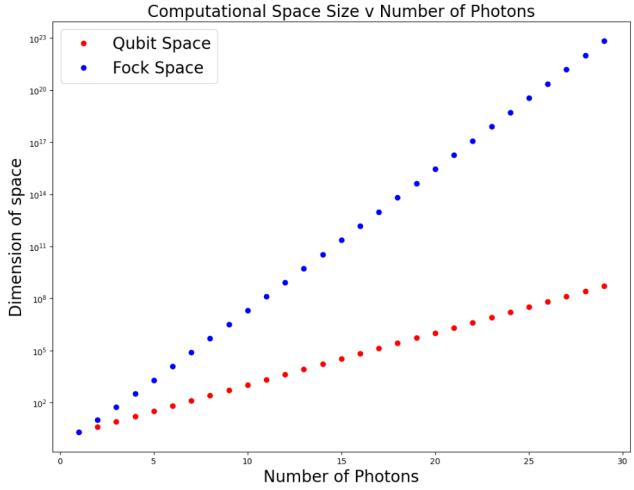


Figure 1: Fock State dimension growth with photons

In Quandela's photonic quantum computers, single photon sources based on quantum dots are used to generate a photon [6]. To generate multiple photons into different modes, an optical demultiplexer is used to deflect a stream of single photons into optical fibres of different lengths [7]. These optical fibres represent waveguides, which define the modes. A universal interferometer is embedded into a photonic chip. This interferometer is capable of implementing a unitary operation (with maximum dimension equal to the total number of modes on the chip. For reference, Quandela's latest photonic quantum computer has 12 modes [7, 8]).

Measurement is then performed by connecting each output mode to a high-efficiency superconducting nanowire single-photon detector (SNSPD) [9]. One important feature of the detection currently unavailable in Quandela's hardware

is the notion of photon number resolving (PNR) detecting, which permits multiple photons to be detected in any given mode [10]. This limitation reduces the dimension of the measureable Fock space, thus PNR is a crucial feature to exploit the potential of photonic computation. Fig. 2 shows the architecural design of Quandela's latest quantum computer, Ascella.

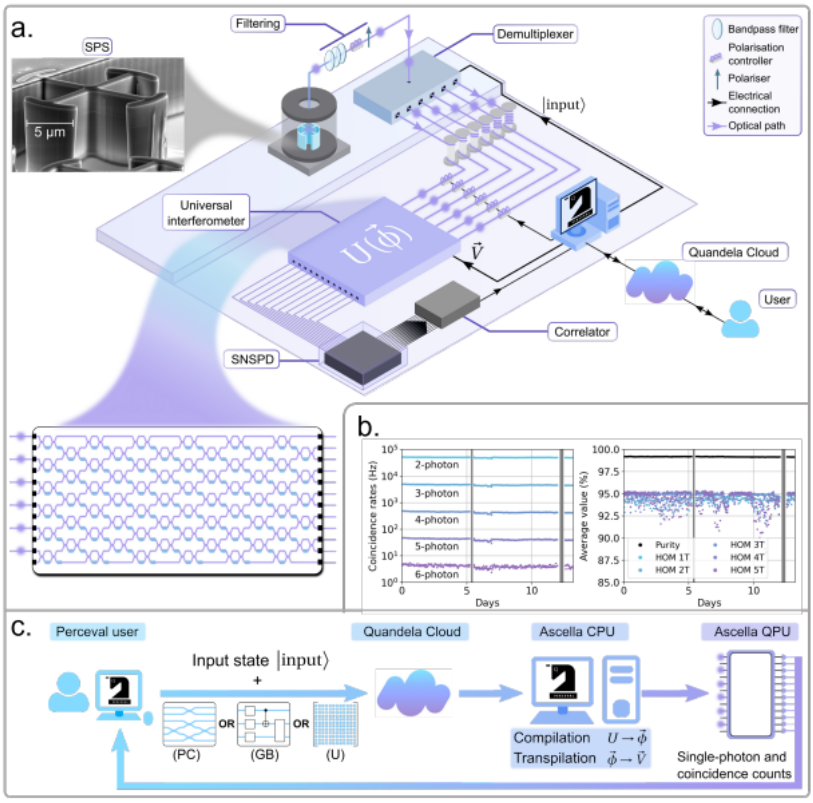


Figure 2: Architecture of Quandela's Ascella Quantum Computer from Maring et al. [7]

2.2 Measurement Based Quantum Computing

Measurement Based Quantum Computing (MBQC) uses the quantum phenomenon of entanglement as a resource by first creating an entangled state, and then using measurement of a subspace to collapse the state, allowing the system to make adaptive decisions based on the measurement outcome. MBQC has been used for applications in error correction [11] and quantum teleportation [12].

A circuit design which uses MBQC can be thought of as a feed forward network, as information is only passed forward as the circuit progresses.

In linear optics, MBQC can be performed by first entangling our photonic state with a generic interferometer, and then detecting photons in a given mode. This measurement will project a reduced state on the rest of the Fock subspace, and each measurement can yield a different component for the adaptive unitary. By repeating this adaptive measurement process on a range of modes, we can create a cascading circuit design by iteratively performing a photonic measurement.

An example of a photonic adaptive circuit is displayed in Fig 3. Suppose we take an initial state of n photons and m modes. We then apply a generic interferometer U_0 across all m modes. This has the effect of entangling the state. The first output mode is then measured using PNR detection. The subsequent unitary U_1 is then dependent on the outcome of this measurement i.e. the circuit adapts to the measurement outcome. This process can be then repeated iteratively. For all $j \in \{1...k\}$, the unitary interferometer U_j, acting on $m-j$ modes, depends on the measurement outcomes p_1,..., p_j. The adaptive measurement outcomes p_1,..., p_k are used as parameters in the computation, whose final outcome is s_1,..., s_{m-k} [13].

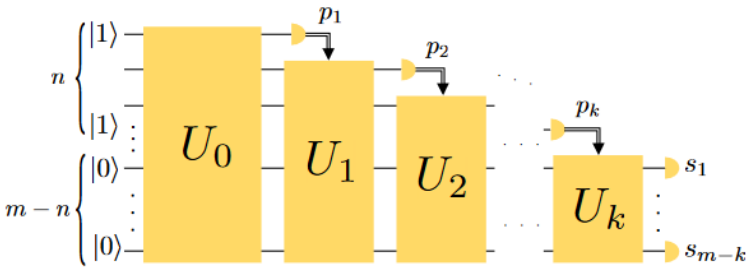


Figure 3: Photonic Adaptive Circuit from Chabaud et al. [13]

2.3 Quantum Machine Learning

Machine Learning refers to the ability of a computer to learn a given task without explicit programming. The role of quantum computation within ML can take different guises, so it is important to clarify what we mean by QML. Firstly, we pose the question, *'What is the model learning?'*. The answer is a solution to a given problem, which is either classical or quantum in nature. Secondly, we can ask *'How is the model learning?'*. This relates to the processor used by the learning model, which can be quantum, classical or a hybrid of both. In this report, I will focus on learning classical problems using a hybrid learning model. The hybrid model consists of using a parameterised quantum circuit to encode the classical data, and then minimising a loss function of the quantum circuit using classical optimisers.

The target task for my QML model takes the form of a Fourier series. This is motivated by two facts. First, a Fourier series can approximate any square-

integrable function over a given interval [14], and also that universal quantum circuits can be used to approximate certain functions [15]. The exponential form for a generic Fourier series is defined in (1). For this report, we consider a truncated Fourier series with integer-valued frequencies, such that $\Omega = \{-K,...,K\}$, with $K \in \mathbf{N}$. K is the *degree* of the frequency spectrum.

$$f(x) = \sum_{n \in \Omega} c_n e^{inx} \tag{1}$$

When comparing the quality of different QML models, it is prudent to ask how we can measure the performance of a given model. This question prompts the definition of three different metrics below.

1. *Expressivity* measures the size of the accessible subspace of a model i.e. it is the set of all unitaries which the model can possibly learn. The greater the ratio of a model's accessible subspace to the set of all unitaries of dimension d, where d is the dimension of the circuit, the more expressive the ML model is [16].

2. *Trainability* measures how 'easily' a model can access an optimal solution. This can reflect the speed of the model (in runtime or total epochs required), the final loss outcome, or the number of computations made.

3. *Generalizability* measures the scope of tasks a model can solve. The larger the set of problems a model can solve, the more generalizable it is.

With these definitions, I conclude the introduction and proceed to take a deeper look at how photonic feed forward networks can be used to model a Fourier series.

3 Method

I begin by exploring a learning using a parameterized photonic circuit consisting of two generic interferometers, with one phase shifter between the interferometers. The phase shifter is used to encode data. For this project, the target function will be univariate, hence I only require one phase shifter per datapoint. Other strategies for encoding classical data into a quantum device have been suggested, such as parallel encoding [17], however a single phase shifter has proved sufficient for the purpose of this project. The circuit (2) is displayed in diagram Fig. 4.

$$U(x, \Theta) = W_2(\theta_2) S(x) W_1(\theta_1) \tag{2}$$

Using an adaptive measurement, I can create a feed-forward circuit by allowing W_2 to be a set of $n+1$ unitaries, each mapped to a different measurement outcome p. Our circuit then becomes (3), and is displayed in Fig. 5.

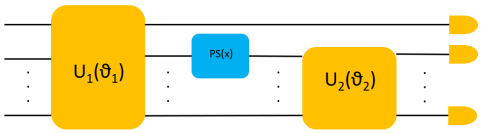


Figure 4: Non-feed forward photonic circuit

$$U(x, \Theta) = W_2^p(\theta_2)S(x)W_1(\theta_1) \tag{3}$$

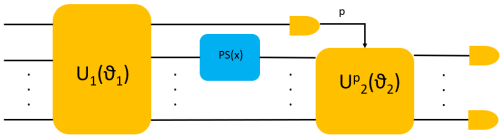


Figure 5: Feed-forward photonic circuit

To calculate the output of the model for a given input state i, I first obtain the expectation value for each output state j. One can take the square of its modulus and multiply by an observable weight λ_j. By summing over each output state, I obtain the below expression (4), where summation runs over the possible different combinations of n photons in the m modes, which equals $\binom{n+m-1}{n}$ possible output states, labelled as d.

$$f(x, \Theta) = \sum_j \lambda_j |\langle j|U(x, \Theta)|i\rangle|^2 \tag{4}$$

I note that the observable weights λ_j can be included in the set of parameters to be optimised in our model, with significant improvement in expressivity. However, this is not required as part of the feed forward analysis and greatly increases the runtime of simulations, hence I use fixed values for the observable weights.

As above, the model's output is dependent on the expectation value of the circuit for given initial state and parameters. hence it is this I explore in greater detail, specifically relating to photonic FF circuits.

3.1 Fourier analysis

For the circuit in Fig. 4, the expectation value of output state $n^{(j)}$ for input state $n^{(i)}$ can be written as equation (5). For brevity, I have dropped the θ parameter from the notation.

$$\langle \boldsymbol{n}^{(j)}|U(x)|\boldsymbol{n}^{(i)}\rangle = \langle \boldsymbol{n}^{(j)}|W_2S(x)W_1|\boldsymbol{n}^{(i)}\rangle, \tag{5}$$

where $|\boldsymbol{n}^{(i)}\rangle = |n_0^{(i)}, n_1^{(i)}, ..., n_m^{(i)}\rangle$.

Now,

$$W_1|\boldsymbol{n}^{(i)}\rangle = \sum_k \alpha_k |\boldsymbol{n}^{(k)}\rangle, \tag{6}$$

where the summation run over the Fock basis set i.e. W_1 creates a superposition of all our possible states, with probability amplitude α_k. Additionally, k runs over the set of Fock states of dimension d. Our data encoding in the phase shifter is proportional to the number of photons k_2 in the second mode [18].

One can write

$$\begin{aligned} S(x)W_1|\boldsymbol{n}^{(i)}\rangle &= \sum_k e^{ik_2 x}\alpha_k |\boldsymbol{n}^{(k)}\rangle, \\ &= \sum_k \beta_k(x)|\boldsymbol{n}^{(k)}\rangle, \\ &= \sum_k \beta_k(x)|n_1^{(k)}\rangle \otimes |n_2^{(k)}, ..., n_m^{(k)}\rangle, \end{aligned} \tag{7}$$

where $\beta_k(x) = \alpha_k e^{n_2^k x}$, and can be considered as a normalized data-encoded coefficient. In the last line, I have split the state into a tensor product of the first mode and the remaining modes, as I note that W_2 is the identity operator on the first mode. This will be useful when I look at the FF case.

I now consider what the output state looks like after applying W_2

$$\begin{aligned} W_2 S(x)W_1|\boldsymbol{n}^{(i)}\rangle &= \sum_k W_2\beta_k(x)|n_1^{(k)}\rangle \otimes |n_2^{(k)}, ..., n_m^{(k)}\rangle, \\ &= \sum_k \beta_k(x)|n_1^{(k)}\rangle \otimes W_2|n_2^{(k)}, ..., n_m^{(k)}\rangle, \end{aligned} \tag{8}$$

Finally, the expectation value is then

$$\langle \boldsymbol{n}^{(j)}|W_2 S(x)W_1|\boldsymbol{n}^{(i)}\rangle = \sum_k \beta_k(x)\langle \boldsymbol{n}^{(j)}|(|n_1^{(k)}\rangle \otimes W_2|n_2^{(k)}, ..., n_m^{(k)}\rangle), \tag{9}$$

This expression is a sum of d terms, and each term is a function of W_2.

To get the final model output, I get the square of the modulus of (9), multiply by the corresponding observable weight, and then sum over all possible output states. This gives

$$f(x) = \sum_{\boldsymbol{n}^{(j)}} \lambda_j |\langle \boldsymbol{n}^{(j)}|W_2 S(x)W_1|\boldsymbol{n}^{(i)}\rangle|^2 \tag{10}$$

Importantly, the above equation is actually a double summation. The outer sum is over all possible output states. The inner sum is over the superposition of all possible states that the circuit processes. Both these sums have d terms, thus the overall output consists of d^2 terms, and each term is a function of W_2.

Alternatively, (10) can be rewritten, by grouping the exponential terms together, and the result is a Fourier series.

$$f^{(n)}(x) = \sum_{\omega \in \Omega_n} c_\omega e^{i\omega x}, \tag{11}$$

where $\Omega_n = \{-n,...,n\}$, and $c^*_\omega = c_{-\omega}$ and n is the number of photons in our model [18].

Now, let's compare this to the FF model in Fig. 5. In this example, W_2 is a parameterized unitary selected from the set of unitaries $W_2^p = \{W_2^{0'},...,W_2^n\}$, where p is the adaptive measurement outcome.

$$\langle \boldsymbol{n}^{(j)}|W_2^p S(x) W_1|\boldsymbol{n}^{(i)}\rangle = \sum_{p=0}^{n} \langle \boldsymbol{n}^{(j)}|(|p\rangle \otimes W_2^p \sum_r \beta_{p,r}(x)|r\rangle), \tag{12}$$

where $|r\rangle = |n_2^{(k)}, ..., n_m^{(k)}\rangle$, and I have relabelled the coefficent as $\beta_{p,r}$. Note that the r summation runs over the basis of the reduced subspace. This makes use of the below result

$$\binom{n+m-1}{n} = \sum_{p=0}^{n} \binom{n-p+m-2}{n-p}, \tag{13}$$

which is easily proved by iteratively reducing the left hand side using the below identity

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}, \tag{14}$$

Looking at (12), I have written the expectation value in the FF case as a sum of terms grouped by the adaptive unitary used. It is then a function of the set of our adaptive unitaries.

Also, if I set $W_2^p = W_2 \ \forall p \in \{0,...,n\}$, the FF case reduces to the non-FF case, with respect to the output probability distribution. This means that the non-FF case is contained within the FF case, which is an important result.

Similarly to the non-FF circuit, the final model output looks like the below

$$f(x) = \sum_{\boldsymbol{n}^{(j)}} \lambda_j |\langle \boldsymbol{n}^{(j)}|W_2^p S(x) W_1|\boldsymbol{n}^{(i)}\rangle|^2 \tag{15}$$

Again, this is a sum of d^2 terms. The difference is the output can be grouped in terms of the dynamic unitaries W_2^p, yielding a function dependent on all of them.

Each Fourier coefficient is then a sum of weighted probability distributions. In the FF case, each distribution is a function of a larger set of unitaries, so it has wider bounds in the range of values it can take under optimization. This gives greater scope to our overall output, thus increasing the expressivity of our model. It becomes a complex optimization problem to find the ideal set of unitaries to best model a given Fourier series, especially as we increase the size of our model, given the exponential growth of d with n and m. Adding more adaptive measurements, as in Fig. 3, will increase the expressivity of our

model when compared to non-FF routines, though more work is required here to understand the scope of the impact of adding more adaptive measurements.

For clarity, I now consider a specific example, and show explicitly how the above approach can be expressed. Suppose our circuit has 3 modes and 3 photons. Our Fock space then has dimension $\binom{5}{3} = 10$. And our set of basis states is as follows

$$V = \{|003\rangle, |012\rangle, |021\rangle, |030\rangle, |102\rangle, |111\rangle, |120\rangle, |201\rangle, |210\rangle, |300\rangle\} \tag{16}$$

For the non-FF case and arbitrary initial state $|i\rangle \in V$, we have

$$U(x)|i\rangle = \sum_k W_2 \beta_k |k\rangle, \tag{17}$$

which is a sum of 10 terms. Our final model output is then a sum of $10^2 = 100$ terms, each a function of W_2.

For the FF case, we have

$$\begin{aligned} U(x)|i\rangle = &|0\rangle \otimes W_2^0(\beta_{003}|03\rangle + \beta_{012}|12\rangle + \beta_{021}|21\rangle + \beta_{030}|30\rangle) \\ &+ |1\rangle \otimes W_2^1(\beta_{102}|02\rangle + \beta_{111}|11\rangle + \beta_{120}|20\rangle) \\ &+ |2\rangle \otimes W_2^2(\beta_{201}|01\rangle + \beta_{210}|10\rangle) \\ &+ |3\rangle \otimes W_2^3 \beta_{300}|00\rangle \end{aligned} \tag{18}$$

This shows how we can group the terms by the adaptive measurement outcome. In this example, the expectation value for a given output state is then a sum, consisting of 4 terms containing W_2^0, 3 terms containing W_2^1, 2 terms containing W_2^2 and 1 term containing W_2^3. Summing over all output states to get the final model output, our Fourier approximation consists of 40 terms containing W_2^0, 30 terms containing W_2^1, 20 terms containing W_2^2 and 10 terms containing W_2^3. Our FF model has greater flexibility in its choices for Θ, though it is a more complex job for the optimizer to find a set of parameters which can give an acceptable approximation.

Also, note that if we set $W_2^p = W_2 \ \forall p \in \{0, 1, 2, 3\}$, the FF case reduces to the non-FF case, with respect to the model's output probability distribution.

3.2 Photonic Simulations

To run simulations of photonic circuits, I have used Quandela's open source software platform Perceval [19]. Currently, there is no in-built adaptive measurement functionality within Perceval, so this presents a challenge in running FF simulations, especially in a QML context where computational cost can result in extremely long run times. To account for this, our circuit makes use of post-selection. The circuit is run for the $n + 1$ values of W_2^p, and only retains the results where W_2^p corresponds to the measurement outcome p of the first mode. Using this probability, alongside the probability of measuring p from the subcircuit, we can use conditional probability to simulate the probability distribution of a FF circuit.

$$\mathbf{P}(p, n_1, ..., n_m) = \mathbf{P}((n_1, ..., n_m)|p)\mathbf{P}(p), \tag{19}$$

where n_j is the number of photons measured in the jth mode.

To embed the target function on to the quantum device, the function domain is discretised into a linear data set, and the circuit is run once per datapoint. Our ansatz is initialised with an input state, and a set of normally distributed random parameters (Θ, λ), and our model Θ optimizes against a loss function for our target Fourier sum. Optimisation is performed using a mean-squared error (MSE) loss function and using the BFGS method. I obtained more accurate results using stochastic gradient descent methods such as basin hopping, though this came at a significant run time cost so they are not included in my results.

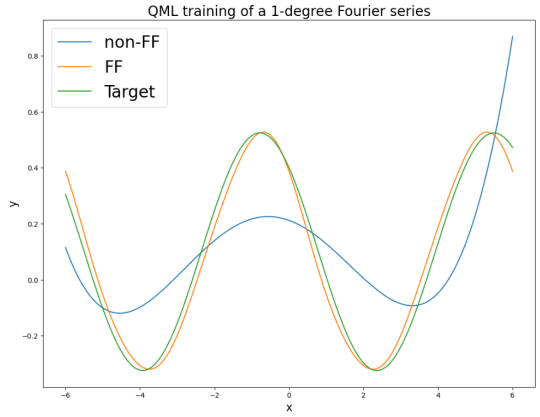
4 Results

For my circuits, I use 3 photons and 3 modes, and my simulations were run against Perceval's Strong Linear Optical Simulation (SLOS) backend, and the the state is initialised with all photons in the first mode i.e. $|i\rangle = |n0...0\rangle$.

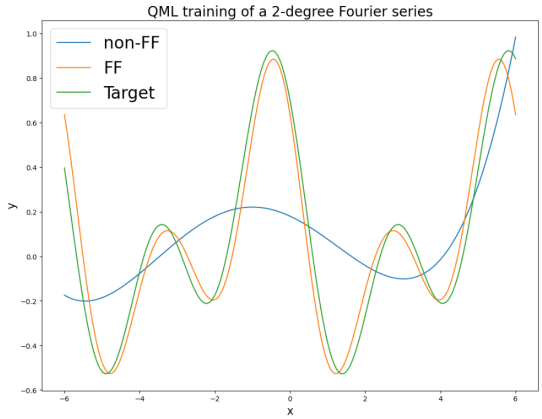
In the below plots, our target Fourier series is of the below form,

$$f(x) = \sum_{\omega=-n}^{n} c_\omega e^{i\omega x}, \tag{20}$$

I set $c_0 = 0.1$ and $c_\omega = 0.15 - 0.15i$ where $\omega > 0$. I run the non-FF and FF circuits described in Fig. 4 and 5 respectively, and measure their accuracy in modelling Fourier series of different degrees.



(a)



(b)

(c)

Figure 6: QML approximations of Fourier series of varying degree

Owing to the increased expressivity of its ansatz, the results show that the FF circuit has a more accurate result in approximating the solution.

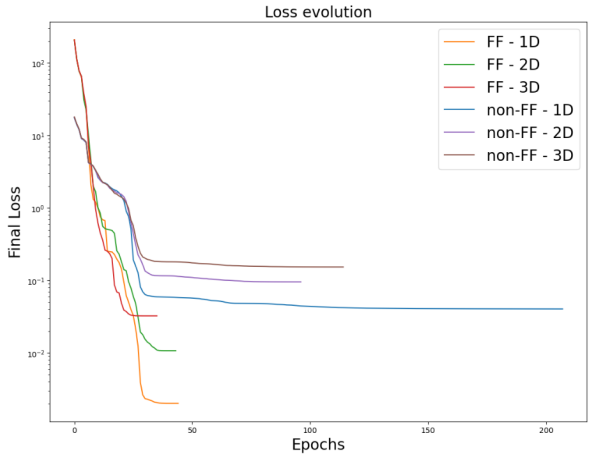


Figure 7: Loss Evolution of the different models

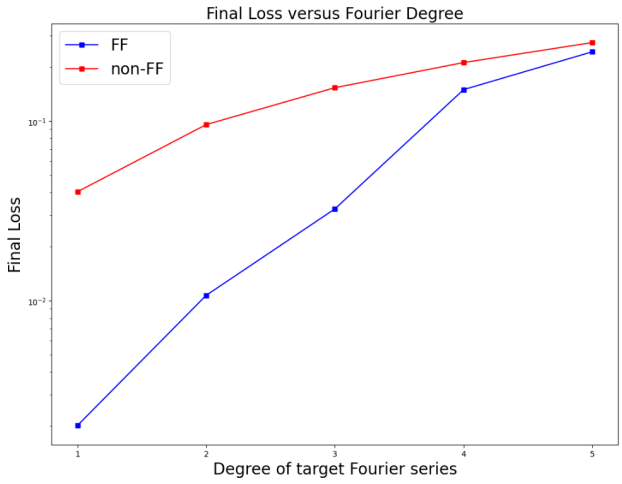


Figure 8: Final loss for different Fourier series

The loss evolution plot in Fig. 7 shows the improved performance of the FF circuits compared to non-FF. It also shows how the model worsens as the degree of the target function increases. This can be explained by the increased complexity of the Fourier series as the degree is increased.

Fig. 8 shows that the benefit of using FF over non-FF reduces as the target function degree is increased. As the non-FF case is contained with FF, it is expected that the FF (blue-line) is at least as good as the non-FF case.

5 Conclusion

My theoretical and numerical analysis present encouraging evidence of the potential of using feed-forward photonic circuits for QML. FF circuits increase the expressivity of a ML model in certain cases. The report leaves open many questions around generalising this use case for arbitrary Fourier series. Gaining further understanding of how to optimise the ansatz used, such as the initial state, initial parameters and circuit architecture, would greatly improve the trainability and expressivity of the model. I am interested in the concept of unitary matrix differentiation, as this is the crux of the ML optimizer, to minimize a multivariate function with respect to a set of circuit unitaries.

Furthermore, it is my ambition to run some of my QML models using FF circuits on Quandela's QPU Ascella during the rest of my internship. I want to express my gratitude to everyone at Quandela who have helped me so much during my stay. A special mention to all on the Applications team and my supervisor Arno Ricou for their assistance and patience.

References

[1] Maria Schuld et al. "Measuring the similarity of graphs with a Gaussian boson sampler". In: *Physical Review A* 101.3 (Mar. 11, 2020), p. 032314.

[2] Hsin-Yuan Huang et al. *Quantum advantage in learning from experiments.* arXiv:2112.00778. arXiv, Dec. 1, 2021.

[3] Alejandro Perdomo-Ortiz et al. "Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers". In: *Quantum Science and Technology* 3.3 (June 19, 2018), p. 030502.

[4] E. Knill, R. Laflamme, and G. J. Milburn. "A scheme for efficient quantum computation with linear optics". In: *Nature* 409.6816 (Jan. 2001). Number: 6816 Publisher: Nature Publishing Group, pp. 46–52.

[5] Federico M. Spedalieri, Hwang Lee, and Jonathan P. Dowling. "High-fidelity linear optical quantum computing with polarization encoding". In: *Physical Review A* 73.1 (Jan. 24, 2006), p. 012334.

[6] N. Somaschi et al. "Near-optimal single-photon sources in the solid state". In: *Nature Photonics* 10.5 (May 2016), pp. 340–345.

[7] Nicolas Maring et al. *A general-purpose single-photon-based quantum computing platform.* June 1, 2023. arXiv: `2306.00874[quant-ph]`.

[8] C Taballione et al. "A 12-mode Universal Photonic Processor for Quantum Information Processing". In: (), p. 11.

[9] Ekkehart Schmidt et al. "Characterization of a photon-number resolving SNSPD using Poissonian and sub-Poissonian light". In: *IEEE Transactions on Applied Superconductivity* 29.5 (Aug. 2019), pp. 1–5.

[10] F Marsili et al. “Physics and application of photon number resolving detectors based on superconducting parallel nanowires”. In: *New Journal of Physics* 11.4 (Apr. 30, 2009), p. 045022.

[11] Stefanie Barz et al. “Demonstrating an element of measurement-based quantum error correction”. In: *Physical Review A* 90.4 (Oct. 2, 2014), p. 042302.

[12] Andrew Childs. *Lecture Notes, Teleportation-based approaches to universal quantum computation with single-qubit measurements.*

[13] Ulysse Chabaud, Damian Markham, and Adel Sohbi. “Quantum machine learning with adaptive linear optics”. In: *Quantum* 5 (July 5, 2021), p. 496.

[14] Lennart Carleson. “On convergence and growth of partial sums of Fourier series”. In: *Acta Mathematica* 116.none (1966), pp. 135–157.

[15] Adrián Pérez-Salinas et al. “Data re-uploading for a universal quantum classifier”. In: *Quantum* 4 (Feb. 6, 2020), p. 226.

[16] Zoë Holmes et al. “Connecting ansatz expressibility to gradient magnitudes and barren plateaus”. In: *PRX Quantum* 3.1 (Jan. 24, 2022), p. 010313.

[17] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. “The effect of data encoding on the expressive power of variational quantum machine learning models”. In: *Physical Review A* 103.3 (Mar. 24, 2021), p. 032430.

[18] Beng Yee Gan, Daniel Leykam, and Dimitris G. Angelakis. “Fock State-enhanced Expressivity of Quantum Machine Learning Models”. In: *EPJ Quantum Technology* 9.1 (Dec. 2022), p. 16. ISSN: 2662-4400, 2196-0763. DOI: 10.1140/epjqt/s40507-022-00135-0. URL: http://arxiv.org/abs/2107.05224.

[19] Nicolas Heurtel et al. *Perceval: A Software Platform for Discrete Variable Photonic Quantum Computing.* arXiv, Apr. 1, 2022. DOI: 10.48550/arXiv.2204.00602.