Clifford Algebras, Quantum Neural Networks and Generalized Quantum Fourier Transform

Marco A. S. Trindade*, Vinícius N. A. Lula-Rocha and S. Floquet

Abstract. We propose models of quantum perceptrons and quantum neural networks based on Clifford algebras. These models are capable to capture geometric features of classical and quantum data as well as producing data entanglement. Due to their representations in terms of Pauli matrices, the Clifford algebras seem to be a natural framework for multidimensional data analysis in a quantum setting. In this context, the implementation of activation functions, and unitary learning rules are discussed. In this scheme, we also provide an algebraic generalization of the quantum Fourier transform containing additional parameters that allow performing quantum machine learning based on variational algorithms. Furthermore, some interesting properties of the generalized quantum Fourier transform have been proved.

Keywords. Clifford algebras, Quantum neural networks, Quantum Fourier Transform.

1. Introduction

Artificial intelligence and machine learning are disciplines that has been created back in the 50’s with the creation of an actual machine called perceptron [26]. This machine was capable to learn some tasks based on the processes of human learning. Nowadays perceptron is a mathematical model comprised of individual units called artificial neurons, activation functions, and weights. These neurons exchange information between the input data and the output via the application of weights and activation functions. On another hand, the process of learning of the perceptron consists of a collection of data one

*Corresponding author.
desires the perceptron to learn, called training set, and a cost function to be minimized via some optimization algorithm in order to update the weights. With the modern advances of classical computing and the improvements of mathematical models, which some are physics-based, perceptron gives place to neural networks, deep neural networks, generative adversarial networks, tensor networks, Boltzmann machines, and so on, that are part of everyday technologies such as pattern recognition [13], medical diagnosis [8,16], combinatorial optimization problems [36].

However, classical computers have been challenged with the volume of data to be processed in some applications that justify the creation of quantum models of perceptrons and neural networks to be implemented in new technologies under development, the quantum computers. In this scenario, we propose Clifford algebra based models of perceptrons and quantum neural networks with unitary learning. Our work adds efforts to many authors who developed unitary quantum models to overcome the issues of the seminal proposals [1,21], which do not have unitary learning rule—unitarity is an important feature for quantum learning. For example, Silva [14] et al. defined a quantum perceptron over a field in order to overcome the limitations of high-cost learning algorithms in classical neural networks. It was proposed a superposition-based architecture learning algorithm (SAL) to optimize the weights of a neural network. In reference [5], Beer et al. introduced quantum deep neural networks with an efficient quantum training algorithm, using the quantum fidelity as a cost function. In this scheme, tolerance to noisy data training has been demonstrated. Another model with a unitary quantum learning rule, based on quaternions, was proposed by E. Bayro-Corrochano [3]. An interesting model was proposed by Shao [28] consists of a quantum feed-forward neural network whose learning algorithm is unitary and it contains quantum superposition and parallelism features. In addition, the Hadamard and swap tests were explored. The procedure is analogous to variational quantum eigensolvers.

Our algebraic proposal is related to variational quantum circuits. Variational quantum circuits, which are also sometimes quantum neural networks can be explored as quantum machine learning models [6,27]. Due to limitations of near-term quantum computing, variational circuits have provided new perspectives that exceed the issues related to computational speedups [17,23] (which are obviously extremely relevant and constitute the gold standard of algorithmic design [27]). In this context, the usefulness of quantum properties such as superposition and entanglement [11] can be investigated [27]. As highlighted in Ref. [7], quantum systems can generate patterns in data that are not feasible for classical systems. It is also important to mention that quantum machine learning may be able to recognize and classify patterns in data that are inaccessible in nowadays classical models designed for classical devices [7]. Possible guides to build more complex architectures capable to capture non-classical features and geometric and topological information are the so-called geometric algebras (Clifford algebras).

Our models of, what we named, Clifford quantum neural networks, explore representations of the Clifford algebras and their relationships with
Hermitian operators in order to obtain suitable representations of the Clifford algebras $Cl(2n,0)$ and $Cl(3,0)^\otimes n$.

We can contextualize our work by invoking several papers whose algebraic approach is fundamental. For instance, a neural network model based on quantum information processing and quaternions has been proposed by Teguri et al. [30] called quaternionic qubit neural network (QQNN). There, the authors showed that numerical experiments indicate better performance in prediction time-series of a chaotic system, compared to conventional real-valued network. Also in the context of quaternions, E. Bayro-Corrochano introduced the quaternion quantum neurocomputing in geometric algebra. Moreover, E. Bayro-Corrochano et al. [3,4] also formulated a quaternionic quantum neural network with applications in pattern recognition. In these works, it was shown that the QQNN has better performance than others approaches, once it requires fewer inputs per pattern (APM- autonomous perceptron models).

In the scope of Clifford algebras applied to general quantum computing, we can mention a few examples as the use of the aforementioned algebra in the design of neural architectures capable to process a plethora of geometric objects [9,10]; the use of Jordan-Wigner representations in Gaussian circuits [20] and the simplification of the computational complexity of the Grover algorithm [2]. In this context, one of us [31] developed a formalism based on Clifford algebras for decoherence-free subspaces, a special class of quantum error correcting codes. Still from an algebraic point of view, it is worth mentioning that symmetry groups have recently been investigated in quantum Boltzmann machines [29].

In this paper, we analyze the underlying algebraic structure of quantum neural networks in the scenario of variational quantum circuits. From this point of view, we propose quantum neural networks based on Clifford algebras $Cl(2n,0)$ and $Cl(3,0)^\otimes n$, and in this context, we propose a Clifford generalization of quantum Fourier transform with parameters to be optimized in our variational models of quantum neural networks.

We organized the paper as follows: in Sect. 2 we present our models of the quantum perceptrons and quantum neural networks. In Sect. 3, we perform error analysis with Hamiltonian simulation techniques for the implementation of our models. Section 4 contains an algebraic generalization of the quantum Fourier transform and its properties. Section 5 is devoted to the conclusions and perspectives. In addition, in Appendix A we review some basic concepts of Clifford algebras. Appendix B contains the swap test and in Appendix C, we provide an example of a simple circuit based on a representation of a Clifford algebra.
2. Clifford Quantum Perceptron and Clifford Quantum Neural Network Models

In this section, we will present two models of quantum perceptrons and two models of quantum neural networks based on Clifford algebras. In order to achieve this objective, it is needed to build unitary operators from a suitable representation of Clifford algebras. In this context, these unitary operators must contain parameters related to the data and weights of our models. A convenient way to obtain the required arbitrary unitary operators is to find Hermitian operators in such representation. More specifically, it is needed to find a full basis of the Hermitian matrices from \( Cl(2n) \), where \( n \) represents the number of qubits of the model. The following lemma shows how to obtain such a basis.

**Lemma 1.** Let \( Cl(2n,0) \equiv Cl(2n) \) be a Clifford algebra of a 2n-dimensional complex vector space endowed with a Euclidean metric. There is a representation of this algebra whose basis induces itself a basis of the vector space of \( 2^n \times 2^n \) Hermitian matrices over \( \mathbb{R} \).

**Proof.** Let’s consider the following representation for the generators of \( Cl(2n) \) [33]

\[
\Gamma_{2k} = I \otimes \cdots \otimes I \otimes \sigma_x \otimes \sigma_z \otimes \cdots \otimes \sigma_z, \\
\Gamma_{2k+1} = I \otimes \cdots \otimes I \otimes \sigma_y \otimes \sigma_z \otimes \cdots \otimes \sigma_z, \tag{1}
\]

with \( k = 0,1,\ldots,n-1 \). These \( 2n \) matrices are linearly independent and Hermitian. Now we consider the \( 2^{2n} \) operators [35]

\[
1, \Gamma_{j_1}, i\Gamma_{j_2}, i\Gamma_{j_3} \Gamma_{j_4}, \ldots, \omega \Gamma_{j_1} \Gamma_{j_2} \cdots \Gamma_{j_s}, \ldots, \omega \Gamma_{j_1} \Gamma_{j_2} \cdots \Gamma_{j_t}, \ldots, \omega \Gamma_{j_1} \Gamma_{j_2} \cdots \Gamma_{2n}, \tag{2}
\]

where \( 0 \leq j_1 \leq j_2 \leq \cdots \leq 2n-1 \), \( \omega = i \) if \( \tilde{j}_1,\ldots,\tilde{j}_c = (-1)^c(-1)^c \Gamma_{j_1},\ldots,\Gamma_{j_c} = (-1)^c(-1)^c \Gamma_{j_1},\ldots,\Gamma_{j_c} = \Gamma_{j_1},\ldots,\Gamma_{j_c} \) and \( i \) is the imaginary unit. The insertion of \( i \) ensures that matrices are Hermitian. Hence using the representation (1) we have \( 2^{2n} \) linearly independent Hermitian matrices, i.e., a basis for the vector space of Hermitian matrices of dimension \( 2^n \times 2^n \). \(\square\)

**Example 1.** For the \( Cl(2) \), we have \( I, \Gamma_0 = \sigma_x, \Gamma_1 = \sigma_y, i\Gamma_0 \Gamma_1 = i\sigma_x \sigma_y \).

**Example 2.** For the \( Cl(2^2) = Cl(4) \), the elements of basis are \( 1, \Gamma_0 = I \otimes \sigma_x, \Gamma_1 = I \otimes \sigma_y, \Gamma_2 = \sigma_x \otimes \sigma_z, \Gamma_3 = \sigma_y \otimes \sigma_z, i\Gamma_0 \Gamma_1 = i(I \otimes \sigma_x \sigma_y), i\Gamma_0 \Gamma_2 = i(I \otimes \sigma_x \sigma_z), i\Gamma_0 \Gamma_3 = i(I \otimes \sigma_y \sigma_z), i\Gamma_0 \Gamma_2 \Gamma_3 = i(I \otimes \sigma_x \sigma_y \otimes I), i\Gamma_1 \Gamma_2 \Gamma_3 = i(I \otimes \sigma_x \sigma_y \otimes \sigma_x \sigma_y). \)

**Remark 1.** Analogous results of Lemma 1 were obtained by Refs. [33,35]. In Ref. [33], it was presented a construction method of universal quantum gates
via Lie and Clifford algebras. Note that in quantum computing the only requirement of a quantum gate is to be a unitary operator acting on the qubit space. In this context, commutation relations were used to generate the Lie algebra $\mathfrak{u}(2^n)$. In another hand, in Ref. [35], Weher and Winter showed that anticommuting observables based on Clifford algebras obey the strongest possible uncertainty relations for the von Neumann entropy and applications in quantum cryptography were discussed. In this work, the authors obtained a real basis for Hermitian operators by observing that the operators presented there form a complete basis with respect to the Hilbert-Schmidt inner product. In contrast with these works, here we use the Clifford algebra reversion operation to build arbitrary Hermitian matrices. In this context, we can obtain unitary operators and relate their parameters with the data and weights of our models.

Elements of Clifford algebras carry a natural geometric interpretation. The scalars and vectors have the standard interpretation. Bivectors can represent an oriented area or an oriented angle associated with rotation and trivectors have an interpretation as an oriented volume. Furthermore, Clifford algebras generalize hypercomplex numbers, including real numbers, complex numbers, and quaternions. These algebraic structures can be used in neural network architectures.

Clifford quantum architectures match entanglement, since the operators are not factorable, with geometric information of data (multivector structure of Clifford algebras). In the next definition, we can encode $2^{2n}$ neurons into $n$ qubits.

**Definition 3.** A Type I Clifford quantum perceptron (CQP-Type I) is defined by $(|x\rangle, |w\rangle, |y\rangle)$ where the input $|x\rangle$ is given by

$$|x\rangle = \exp \left( i \sum_{j=0}^{2^n-1} \omega_j \alpha_j \Gamma^{(V)}_j \right) |0\rangle^\otimes n,$$

(3)

The weight is

$$|w\rangle = \exp \left( i \sum_{j=0}^{2^n-1} \omega_j \theta_j \Gamma^{(V)}_j \right) |0\rangle^\otimes n$$

(4)

and the output is defined as

$$|y\rangle = \exp(i \omega_{\mu} \phi \Gamma^{(V)}_{\mu}) |0\rangle^\otimes n.$$ 

(5)

with $\phi = \arccos(\varphi(|\langle x|w\rangle|))$, where $\varphi$ is the activation function, $\Gamma^{(V)}_{\mu}$ stands for an arbitrary element of the basis for the Clifford algebra $\text{Cl}(2n)$ and $\omega_{\mu}$ is defined in the Lemma 1. It is also convenient to highlight that the $(V)$ superscript notation on $\Gamma^{(V)}_{\mu}$ stands for “vector space”, since we are considering an arbitrary element of the basis of the vector space associated with the algebra and the index $\mu$ refers to a particular element of this basis.

Note that the gates associated with the exponential of a general Hermitian Clifford algebra elements in (3–5) can generate entangled states since they
are not in general factorable. This fact makes the implementation of a circuit to perform such exponentials in quantum computers a difficult task because real quantum computers cannot implement operations in several qubits at once. It is instead required to factorable operations on many qubits into a bunch of operations on a few qubits. In addition, since arbitrary Clifford algebra elements, in general, do not commute, some approximation has to be done in order to factor (3–5) and translate them into a feasible quantum circuit for actual quantum devices. A simplification of the CQP-Type I model can be performed from a selection of some basis elements, such as generators ($2^n$ elements), to compose the unitary operations. A possible choice for polynomial growth is given by the following definition, which is a particular case of Definition 3.

**Definition 4.** A Type II Clifford quantum perceptron (CQP-Type II) is defined by the triple ($|x\rangle, |w\rangle, |y\rangle$), where the input $|x\rangle$, the weight $|w\rangle$ and the output $|y\rangle$ are respectively given by

$$
|x\rangle = \exp \left( i \sum_{j=0}^{2n-1} \alpha_j \Gamma_j \right) |0\rangle^\otimes n, \\
|w\rangle = \exp \left( i \sum_{j=0}^{2n-1} \theta_j \Gamma_j \right) |0\rangle^\otimes n, \\
|y\rangle = \exp(i\phi \Gamma_{\mu})|0\rangle^\otimes n,
$$

with $\phi = \arccos(\varphi(|\langle x|w\rangle|))$, where $\varphi$ is the activation function and $\Gamma_{\mu}$ is an arbitrary generator of the Clifford algebra $Cl(2n)$. The index $\mu$ refers to a particular element of the algebra generators.

Here we encode $2n$ neurons into $n$ qubits so far. However, this scheme can be generalized into a multilayer case, i.e. a multilayer Clifford quantum neural network (MCQNN). In order to perform such a generalization, let’s consider the parameter $\theta_{m,j_m}$, where the index $m$ refers to the $m$th-layer and $j_m$ refers to the $m$th-layer neuron indexes and define

$$
|x\rangle_{m-1} = \exp \left( i \sum_{j=0}^{2n-1} \alpha_{j,m-1} \Gamma_j \right) |0\rangle^\otimes n, \\
|w\rangle_{m,j_m} = \exp \left( i \sum_{j=0}^{2n-1} \theta_{m,j_m} \Gamma_j \right) |0\rangle^\otimes n, \\
|y\rangle_m = \exp \left( i \sum_{j=0}^{2n-1} \phi_{j,m} \Gamma_j \right) |0\rangle^\otimes n.
$$

where $\phi_{j,m} = \arccos(\varphi(|\langle x|w\rangle_{m,j_m}|))$.

For the learning algorithm relative with the CQP-Type II model, we will consider the training set $\{|x^{(s)}\rangle, |r^{(s)}\rangle\}_{s \in S}$ where $|x^{(s)}\rangle$ and $|r^{(s)}\rangle$ are
respectively the input and the desired output of the MCQNN, given by

$$|x^{(s)}\rangle = \exp \left(i \sum_{j=0}^{2n} \alpha_j^{(s)} \Gamma_j \right) |0\rangle^\otimes n, \quad (12)$$

$$|r^{(s)}\rangle = \exp \left(i \beta^{(s)} \Gamma_\mu \right) |0\rangle^\otimes n. \quad (13)$$

For each input $|x^{(s)}\rangle$ the multilayer Clifford quantum neural network returns an output $|y^{(s)}\rangle$ given by

$$|y^{(s)}\rangle = \exp \left(i \phi^{(s)} \Gamma_\mu \right) |0\rangle^\otimes n, \quad (14)$$

where $\phi^{(s)} = \arccos \varphi(|\langle x^{(s)}|w\rangle|)$.

Now we will consider the quantum fidelity between the actual output of the MCQNN and $|y^{(s)}\rangle$ and its respective training vector $|x^{(s)}\rangle$. The quantum fidelity $F^{(s)}$ relative with the sth element of the training set is given by

$$F^{(s)} = |\langle r^{(s)}|y^{(s)}\rangle|$$

$$= |\langle 0| (\cos \phi^{(s)} I - i \sin \phi^{(s)} \Gamma_\mu) (\cos \beta^{(s)} I - i \sin \beta^{(s)} \Gamma_\mu) |0\rangle^\otimes n$$

$$= \cos \phi^{(s)} \cos \beta^{(s)} - \sin \phi^{(s)} \sin \beta^{(s)}, \quad (15)$$

where we used the fact that $\Gamma_\mu^2 = I$.

In this context, the learning rule consists of the optimization, in a classical computer, of the fidelity $F^{(s)}$ relative to the parameters $\theta_j(k)$ of the weight vector of the kth iteration $|w(k)\rangle$ via gradient ascent given by [28]

$$\theta_j(k+1) = \theta_j(k) + \eta \frac{\partial F^{(s)}(k)}{\partial \theta_j}, \quad (16)$$

which results in the weight vector $|w(k+1)\rangle$ associated with the $k + 1$th iteration, given by

$$|w(k+1)\rangle = \exp \left[ i \sum_{j=0}^{2n} \left( \theta_j(k) + \eta \frac{\partial F^{(s)}(k)}{\partial \theta_j} \right) \Gamma_j \right] |0\rangle^\otimes n, \quad (17)$$

where $\eta$ is the learning-rate parameter of the back-propagation algorithm.

**Remark 2.** Shao presents a similar model with vectors $|x\rangle$, $|w\rangle$, $|y\rangle$ representing the data, the weights and the output, respectively. However in his model, the information is stored in a single-qubit state: parameterized rotations $R(\alpha_i)$ $R(\theta_j)$ and $R(\phi)$ are applied in each the last qubit, initially in $|0\rangle$, of each parcel of a n-qubit superposition to achieve the vectors $|x\rangle$, $|w\rangle$, $|y\rangle$, i.e.,

$|x\rangle = \sum_{i=0}^{n-1} \lambda_i |i\rangle |x_i \rangle$, $|w\rangle = \sum_{i=1}^{n-1} \mu_i |i\rangle |w_i \rangle$, where $|x_i \rangle = R(\alpha_i) |0\rangle$, $|w_j \rangle = R(\theta_j) |0\rangle$ and $|y\rangle = R(\phi) |0\rangle$. It is worth noting that the aforementioned rotations cannot generate entangled states since they act only on single-qubit states. The main difference between Shao’s and our work is that we consider unitary operators acting on all n-qubits $|0\rangle^\otimes n$ at once, rather than in single-qubits, allowing the generation of entangled states. This fact can be explained by noting that our unitary operations consist of exponentials of arbitrary Hermitian Clifford algebra element representation which are in
general not factorable. Furthermore, our arbitrary unitary operators contain Clifford multivectors elements which can be associated with multidimensional data and can be used in quantum information. Another remarkable difference between his and our works concerns the learning algorithm. In [28], the learning rule of the parameters is responsible to update the rotation operators that act only on a single qubit. Here, the unitary operator acts on all qubits of the quantum computer processor, creating correlations between the parameters.

Importantly, we can determine \(|\langle \psi|\phi \rangle|\) in a quantum computer by the use of the swap test. For the sake of clarity, we will transfer this discussion to Appendix B.

It is worth noting that the set of CQP-Type II perceptrons is a subset of the set of CQP-Type I perceptrons, since by definition the Type II considers only the basis elements that generate 1-vectors of the Clifford algebras, while Type I consider the basis of all multivectors the algebra. Now we will define and discuss the equivalence between the two types of perceptrons through Theorema 7.

**Definition 5.** Two Clifford quantum perceptrons \((|x\rangle, |w\rangle, |y\rangle)-Type I and \((|x\rangle, |w\rangle, |y\rangle)-Type II are called equivalent if \(|y\rangle = |y\rangle\).

**Definition 6.** A unitary transformation \(U\) acting on CQP is given by \(U|x\rangle\) and \(U|w\rangle\).

**Theorem 7.** Let \((|x\rangle, |w\rangle, |y\rangle\)) and \((|x\rangle, |w\rangle, |y\rangle\)) be two Clifford quantum perceptrons of Types I and II, respectively. Then every unitary transformation \(U\) acting on CQP-Type II produces an equivalent CQP-Type I.

**Proof.** First, let \(U\) be a unitary transformation such that
\[
U|x\rangle = |x^\prime\rangle, \\
U|w\rangle = |w^\prime\rangle.
\]

We have that
\[
\phi^\prime = \arccos \varphi(|\langle x^\prime|w^\prime \rangle|) = \arccos \varphi(|\langle x|U^\dagger U|w \rangle|) = \arccos \varphi(|\langle x|w \rangle|) = \phi.
\]

Since \(|y\rangle\) is uniquely determined by \(\phi\), then \(|y^\prime\rangle = |y\rangle\). However, we need to verify if the conditions of Definition 3 are satisfied. Notice that
\[
|x^\prime\rangle = U \exp \left( i \sum_{j=0}^{2n-1} \alpha_j \Gamma_j \right) |0\rangle^{\otimes n} = U U'|0\rangle^{\otimes n} = U''|0\rangle^{\otimes n},
\]

since \(U'\) is unitary\(^1\) by Lemma 1. Again, by Lemma 1, the new operator \(U''\) can be expressed by \(U'' = \exp \left( i \sum_{j=0}^{2n-1} \omega_j \alpha_j^* \Gamma_j^{(V)} \right)\) and therefore is unitary.

We can proceed analogously for the \(|w^\prime\rangle\) and \(|y^\prime\rangle\) to finish the proof. \(\square\)

---

\(^1\)If a operator \(A\) is Hermitian, then \(\exp(iA)\) is unitary.
Remark 3. Note that is not reciprocal, i.e., a unitary $U$ acting on QCP-Type I does not produce an equivalent QCP-Type II.

It is important to note that one of the best features of the models of Clifford quantum perceptrons Type I and II is to allow the computation of arbitrary activation functions.

In the following, we will propose another two models of Clifford quantum network that allow the implementation of arbitrary activation functions in terms of functions of operators. The main difference between the previous models is that both models in the following enable obtaining the value of the derivatives relative to the process of minimization of the cost function to be measured in a quantum computer. This feature allows us to reduce the dependence of the models with classical computations which is a potential of a speed-up compared to the usual models in the literature.

Definition 8. A variational Clifford quantum neural network Type-I (VCQNN-Type I) is given by unitary operators $U^{\text{Cliff}}(x), U^{\text{Cliff}}(\theta)$, an initial $n$-qubit state $|0\rangle^\otimes n$, a Hermitian operator $A$, a real activation function $\phi$, and an output $\langle x; \theta | \phi(A) | x; \theta \rangle$ defined as

$$|x\rangle = U^{\text{Cliff}}(\theta)U^{\text{Cliff}}(x)|0\rangle^\otimes n = \sum_{i=1}^{n} a_{i;\theta} |x_i\rangle,$$

$$A = \sum_{j} |a_{j;\theta}| |x_j\rangle \langle x_j|,$$

and

$$\phi(A) = \sum_{j} \phi(|a_{j;\theta}|) |x_j\rangle \langle x_j|,$$

with $\sum_{i} |a_{i}|^2 = 1$.

Remark 4. Note that the function $\phi(A)$ is also a Hermitian operator. It is also important to highlight that the superscript index “Cliff” in the unitary operators $U^{\text{Cliff}}$ means they are constructed in terms of Clifford algebra elements. This characteristic allows geometric properties to be carried by the quantum system.

Remark 5. The minimization of an arbitrary cost function must contain a derivative of $\langle A \rangle = \langle x; \theta | \phi(A) | x; \theta \rangle$. For this model to be more easily implemented on a near-term quantum computer, we must consider $\phi(A) = A$, where $A$ is a Hermitian and unitary operator, which can be obtained easily with a basis element, $\Gamma_{\mu}^{V}$, of the representation of the $Cl(2n)$. In particular for $U^{\text{Cliff}}(\theta)U^{\text{Cliff}}(x) = \prod_{i=1}^{n} U(\theta_i) \prod_{i=1}^{n} U(x_i)$, where $U(x_k) = e^{i\alpha_k \Gamma_k}$, $U(\theta_k) = e^{i\theta_k \Gamma_k}$, we have

$$\langle A \rangle = \otimes_{n} |0\rangle U^{\dagger}(x_1) \cdots U^{\dagger}(x_n)U^{\dagger}(\theta_1) \cdots U^{\dagger}(\theta_n)|\phi(A)|U(\theta_1) \cdots U(\theta_n) \cdots U(x_n) \cdots U(x_1)|\otimes_{n}$$
Consequently, the derivative of the expectation value of $A$ in relation to $\theta_k$ is
\[
\frac{\partial \langle A \rangle}{\partial \theta_k} = \otimes_n \langle 0| U^\dagger(x_1) \cdots U^\dagger(x_n) U^\dagger(\theta_1) \cdots U^\dagger(\theta_n) |\phi(A)| U(\theta_1) \cdots U(\theta_n) |x_n\rangle_U \otimes_n \langle x_1| U(\theta_1) \cdots U(\theta_n) |x_n\rangle_U \otimes_n \langle 0| \cdots \otimes_n \langle 0| \cdots \otimes_n \langle 0| U^\dagger(x_1) \cdots U^\dagger(x_n) U^\dagger(\theta_1) \cdots U^\dagger(\theta_n) |\phi(A)| U(\theta_1) \cdots U(\theta_n) |x_n\rangle_U \otimes_n |x_1\rangle_U \cdots \otimes_n |x_n\rangle_U.
\]
Since $\Gamma_k$ is unitary, the two parcels of (24) can be understood as an inner product of two vectors in Hilbert space and therefore can be estimated on a quantum computer by using swap test or Hadamard test.

Another possibility is given by

**Definition 9.** A variational Clifford quantum neural network Type II (VCQNN-Type II) is given by unitary operators $U^{\text{Cliff}}(x)$, $U^{\text{Cliff}}(\{y_d\})$, $U^{\text{Cliff}}(\theta)$, initial state $|0\rangle_U \otimes_n$, the output state $|y_{out}\rangle$ and the training set of desired states $\{|y_{out,d}\rangle\}$, a Hermitian operator $A = \sum_j |a_j(x_j; \theta)||x_j\rangle\langle x_j|$, a real activation function $\phi$ and an output $|y_{out}\rangle$, defined as
\[
|x\rangle = U^{\text{Cliff}}(\theta)U^{\text{Cliff}}(x)|0\rangle_U \otimes_n = \sum_{i=1}^n a_{i,\theta} |x_i\rangle, \tag{25}
\]
\[
A = \sum_j |a_j(x_j; \theta)||x_j\rangle\langle x_j|, \tag{26}
\]
\[
\phi(A) = \sum_j \phi(|a_j(x_j; \theta)||x_j\rangle\langle x_j|, \tag{27}
\]
\[
|y_{out}\rangle = \exp(i\phi(A))|x\rangle = \exp \left( i \sum_j \phi(|a_j(x_j; \theta)||x_j\rangle\langle x_j| \right) |x\rangle, \tag{28}
\]
\[
|y_{out,d}\rangle = U^{\text{Cliff}}(y_d)|0\rangle_U \otimes_n, \tag{29}
\]
where $\sum_i |a_{i,\theta}|^2 = 1$.

**Remark 6.** It is important to highlight that $U^{\text{Cliff}}(\{y_d\})$ is a unitary transformation parameterized by parameters set $\{y_d\}$ that each element $y_d$ correspond to a rotation that implements a desired output $|y_{out,d}\rangle$ of the training set. In another words, each parameter $y_d$ corresponds to a desired output $|y_{out,d}\rangle = U^{\text{Cliff}}(y_d)|0\rangle_U \otimes_n$.

**Remark 7.** Analogously to Remark 5, we must consider $\phi(A) = A$, where $A$ is a Hermitian and unitary operator. In this scheme, we may use the quantum fidelity $F^{(d)} = \langle y_{out}(\theta)|y_{out,d}\rangle$ as cost function. Again if $U^{\text{Cliff}}(x) = \prod_{i=1}^n U(x_i)$, $U^{\text{Cliff}}(\theta) = \prod_{i=1}^n U(\theta_i)$, $U(x_k) = e^{i\alpha_k \Gamma_k}$, $U(\theta_k) = e^{i\theta_k \Gamma_k}$ and $\phi(k)$ unitary, it would be more easily implemented on a near-term quantum computer. The derivative of fidelity is given by
\[
\frac{\partial F^{(d)}}{\partial \theta_k} = \otimes_n \langle 0| U^\dagger(x_1) \cdots U^\dagger(x_n) U^\dagger(\theta_1) \cdots i\Gamma_k U(\theta_k) \cdots U^\dagger(\theta_n) U^{\text{Cliff}}(y)|0\rangle_U \otimes_n,
\]
which, analogously as in the previous definition, can be thought of as an inner product of two vectors of Hilbert space and therefore can be estimated in a quantum computer by swap test.

All these proposals for quantum neural networks based on Clifford algebras may be trained by classical optimization algorithms as in several other proposals for hybrid algorithms [27]. A simple example of a circuit that implements unitary operations is given in Appendix B. It is important to mention that our models can be useful for quantum machine learning of quantum systems since an arbitrary Hamiltonian operator can be written in terms of the representation (1) in according to Lemma 1.

Alternatively, we can build the Hermitian operators from a basis of the space \( \Lambda^0 \mathbb{R}^{3,0} \oplus \Lambda^1 \mathbb{R}^{3,0} \otimes n \) that corresponds to a subspace of the tensor product of Clifford algebras \( Cl(3,0)^{\otimes n} \).

**Proposition 10.** The subspace \( \Lambda^0 \mathbb{R}^{3,0} \oplus \Lambda^1 \mathbb{R}^{3,0} \otimes n \subset Cl(3,0)^{\otimes n} \) is isomorphic to the space vector of Hermitian matrices \( 2^n \times 2^n \).

**Proof.** A basis of subspace \( \Lambda^0 \mathbb{R}^{3,0} \oplus \Lambda^1 \mathbb{R}^{3,0} \otimes n \subset Cl(3,0)^{\otimes n} \) is given by \( \{1, \gamma_1, \gamma_2, \gamma_3\}^{\otimes n} \), for which the elements have a representation in terms of matrices \( \{I, \sigma_x, \sigma_y, \sigma_z\}^{\otimes n} \). Given a vector space \( V \) of dimension \( 2^{2n} \), every linearly independent subset of \( 2^{2n} \) elements is a basis of \( V \). The \( 2^{2n} \) elements are linearly independent and Hermitian matrices. Therefore they form a basis of space of Hermitian matrices \( 2^n \times 2^n \). \( \square \)

We will show in the next section that a formulation based on the tensor product of Clifford algebras \( Cl(3,0)^{\otimes n} \) is associated with a proposal of generalization of the quantum Fourier transform, which can be used for our proposal of quantum neural networks.

### 3. A Way in Direction to Implementation and Error Analysis

One of the pillars of our models defined in the previous section is the unitary operation built as an exponential of a general Hermitian Clifford elements to be applied on a \( n \)-qubit at once. Since actual quantum devices do not perform such operation, it is required to factor this operation in many sequential operations on few qubits. Similar issues are found in the area of Hamiltonian simulation since the aim of this area is to simulate the time evolution of quantum systems governed by some Hamiltonian \( H \). This can be done by the application of the time evolution operator \( e^{iHt} \), where \( t \) is the time parameter, on a \( n \)-qubit state \( |\psi\rangle \).

There are several algorithms for Hamiltonian simulation. We can apply these techniques in our proposal, especially related to Clifford quantum perceptron. Particularly, we analyze the product formula approach [12]. The exponential of a sum of operators is approximated by a product of exponentials. For the \( k \)-local Hamiltonians [25] (a sum of \( L \) Hermitian terms acting upon at most \( k \) qubits), \( L \) is upper bounded by a polynomial in \( n \) and we get better first-order bounds [12]. So we have the following proposition.
Proposition 11. Let $M$ be a POVM element related to the measurement. Let also $P_U$ and $P_V$ be the probabilities of obtaining the associated measurement outcome if the operation $U$ (or $V$), where $U = \exp \left( -it \sum_{j=1}^{2n} \omega_j \eta_j \Gamma_j^V \right)$ and $V = \left[ \prod_{j=1}^{2n} \exp \left( -it \omega_j \eta_j \Gamma_j^V \right) \right]^r$, was performed. Then $|P_U - P_V| \leq 2E(U, V)$, with

$$E \equiv \max_{\langle \psi \rangle} ||(U - V)\langle \psi \rangle||$$

and $\Lambda = \max_j ||H_j||$. We can rewrite the inequality (30). Let $H_j$ be a Hermitian operator [12]. In our case $H_j = \eta_j \omega_j \Gamma_j^V$ and $\Gamma_j^V$ is unitary so that $\Lambda = \max_j |\eta_j|$. Hence

$$|P_U - P_V| \leq 2 \left\| \exp \left( -it \sum_{j=0}^{2n-1} \omega_j \eta_j \Gamma_j^V \right) - \left[ \prod_{j=0}^{2n-1} \exp \left( -it \omega_j \eta_j \Gamma_j^V \right) \right]^r \right\|$$

$$\leq 2 \left( \frac{2(2n \max_j |\eta_j|)^2}{r} \right) \exp \left( \frac{2n \max_j |\eta_j||t|}{r} \right) \text{.}$$

Remark 8. Note that the choice $H_j = \omega \eta_j \Gamma_j^V$ is the worst scenario relative to the analytic error bound $\left( \frac{2(2n \max_j |\eta_j|)^2}{r} \right)$ for the first-order product formula. The definition of matrix multiplication gives a complexity $\Theta(2^{3n})$. However, with $r = 2n$, we obtain $\Theta(2n)$ for each operation in the CQP-Type II.

Proposition 12. The number of pairs of non-commuting elements of the basis of Clifford algebra $Cl(2n, 0)$ is given by

$$\Omega = \sum_{p,q:p<q}^{2n} \#_{p,q} \left( \begin{pmatrix} 2n \\ p \end{pmatrix} \begin{pmatrix} 2n \\ q \end{pmatrix} \right) \text{,}$$

where $\#_{p,q}$ stands for the number of elements for which in the pairs $(p,q)$ $(p$-vector and $q$-vector spaces) we have, in the cases (a)(odd; even), (even; odd), or (even; even) - an odd number of generators appearing simultaneously in the $p$-vector and $q$-vector; (b)(odd; odd) - an even number of generators appearing simultaneously in the $p$-vector and $q$-vector. If no generators are appearing simultaneously in the $p$-vector and $q$-vector, then we have non-commuting elements only in the case that $p$ and $q$ are both odd.

Proof. The number of generators in a $p$-vector is given by $(\#_{p,q}^\text{gen})$, where $\#_{p,q}^\text{gen}$ stands for “number of generators”, so that we must compute all possible combinations of $p$-vector with $q$-vector:

$$\sum_{p=1}^{2n} \sum_{q=1}^{2n} \left( \begin{pmatrix} \#_{p,q}^\text{gen} \\ p \end{pmatrix} \begin{pmatrix} \#_{p,q}^\text{gen} \\ q \end{pmatrix} \right)$$
\[
\begin{align*}
= \binom{2n}{1} \binom{2n}{1} + \binom{2n}{1} \binom{2n}{2} + \cdots + \binom{2n}{1} \binom{2n}{2n} \\
+ \binom{2n}{2} \binom{2n}{1} + \binom{2n}{2} \binom{2n}{2} + \cdots + \binom{2n}{2} \binom{2n}{2n} \\
\vdots \\
+ \binom{2n}{2n} \binom{2n}{1} + \binom{2n}{2n} \binom{2n}{2} + \cdots + \binom{2n}{2n} \binom{2n}{2n}.
\end{align*}
\]

(33)

Most of these terms appear repeatedly so we should only count them once. In order for the elements of the basis not commute, we need an odd number of anti-commutations of the generators in the product of a \(p\)-vector and a \(q\)-vector, \((\gamma_{i_1}, \gamma_{i_2}, \ldots, \gamma_{i_p})(\gamma_{j_1}, \gamma_{j_2}, \ldots, \gamma_{j_q})\). Then we can perform

\[
\gamma_{i_k}(\gamma_{j_1}, \gamma_{j_2}, \ldots, \gamma_{j_q}) = -\langle \gamma_{j_1}, \gamma_{j_2}, \ldots, \gamma_{j_q} \rangle \gamma_{i_k}
\]

(34)
an odd number of times, where \(\gamma_{i_k}\) and \(\gamma_{j_{k'}}\) are constituting generators of the \(p\)-vector \(\gamma_{i_1}, \gamma_{i_2}, \ldots, \gamma_{i_p}\) and of the \(q\)-vector \(\gamma_{j_1}, \gamma_{j_2}, \ldots, \gamma_{j_q}\), respectively. This occurs when we have an odd number of generators \(\gamma_{i_k}\) (which do not appear simultaneously in the \(q\)-vector) and an odd number of generators \(\gamma_{j_{k'}}\).

Consider the case (even; odd) and suppose initially that we have an even number of generators that appear simultaneously in both the \(p\)-vector and \(q\)-vector. Consequently, we have (i) an even number of generators of the \(p\)-vector (which appear simultaneously in the multivectors) and (ii) another even number of generators of the \(p\)-vector (which do not appear simultaneously in the multivectors) that both (cases (i) and (ii)) anti-commute with an odd number of generators of the \(q\)-vector. So we have an even total number of anti-commutations and therefore the basis elements \(\gamma_{i_1}, \gamma_{i_2}, \ldots, \gamma_{i_p}\) and \(\gamma_{j_1}, \gamma_{j_2}, \ldots, \gamma_{j_q}\) do commute.

Consider now an odd number of generators that appear simultaneously in both the \(p\)-vector and \(q\)-vector. So we have (I) an odd number of generators of the \(p\)-vectors (which appear simultaneously in the multivectors) that anti-commute with an even number of generators of the \(q\)-vector and (II) another odd number of generators of the \(p\)-vectors that anti-commute with an odd number of generators of the \(q\)-vectors. Consequently, we have an odd total number of anti-commutations so that basis elements \(\gamma_{i_1}, \gamma_{i_2}, \ldots, \gamma_{i_p}\) and \(\gamma_{j_1}, \gamma_{j_2}, \ldots, \gamma_{j_q}\) do anti-commute. The development is similar for the other cases.

A corollary of this proposition is the commutator error bounds of our models since it depends on the counting of (anti)commutating of multivectors formed by the algebraic products of the generators \(\Gamma_i\) of Lemma 1. The error bound obtained by \[12\] (Theorem F.5) of the first-order product formula can be applied to the context of our models, giving the inequalities:

\[
\left| \exp \left( -it \sum_{j=0}^{2^n-1} \omega_j \eta_j \Gamma_j^{(V)} \right) - \prod_{j=0}^{2^n-1} \exp \left( -it \omega_j \eta_j \Gamma_j^{(V)} \right) \right|^r \leq \Omega \left( \frac{\max_j |\eta_j| t^2}{r} + \frac{(2^n |t|^3 \max_j |\eta_j|)^2}{3r^2} \right) \exp \left( \frac{2^{2n} \max_j |\eta_j| |t|}{r} \right),
\]

(35)
for the CQP-Type I, where \( \Omega \) is defined by (32). In the case where we consider only the algebra generators, i.e., 1-vectors, as in the model the CQP-Type II, we have:

\[
\left\| \exp \left( -it \sum_{j=0}^{2n-1} \eta_j \Gamma_j \right) - \prod_{j=0}^{2n-1} \exp \left( -it \eta_j \Gamma_j \right) \right\| \leq 2n \left( \frac{\max_j |\eta_j||t|}{r} \right)^2 + \frac{(2n|t|^3 \max_j |\eta_j|)^2}{3r^2} \exp \left( \frac{2n \max_j |\eta_j||t|}{r} \right). \quad (36)
\]

4. Generalized Quantum Fourier Transform

We will now present a new quantum Fourier transform based on Clifford algebras that generalizes the usual case. Unlike the standard quantum Fourier transform, this generalization contains parameters that can be used to model quantum neural networks as in the previous section. However, its applications can go beyond. There are some approaches to the classical Fourier transform related to Clifford algebras. An excellent review can be found in reference [15].

Here we are interested in a generalization of quantum Fourier transform to be used in our models of quantum neural networks based on generators of Clifford algebras. In other words, this generalization must be unitary and factorable into unitary operators to be applied in single-qubits in order to be possible to build a quantum circuit that can exactly implement it. A good alternative is to consider the elements \( \Gamma_k \) formed by the Kronecker sum of Pauli representations of 1-vectors of the algebra \( Cl(3,0) \), i.e,

\[
\Gamma_k = (\alpha_{x_1}^{k_1} \sigma_x + \alpha_{y_1}^{k_1} \sigma_y + \alpha_{z_1}^{k_1} \sigma_z) \otimes \cdots \otimes I + \cdots + I \otimes \cdots \otimes (\alpha_{x_n}^{k_n} \sigma_x + \alpha_{y_n}^{k_n} \sigma_y + \alpha_{z_n}^{k_n} \sigma_z), \quad (37)
\]

where \((\alpha_{x_1}^{k_1}, \alpha_{y_1}^{k_1}, \alpha_{z_1}^{k_1}) \in \mathbb{R}^3, 1 \leq i, \ell \leq n, \) is a unit vector. Each \( k_\ell \) is a element of the binary index set \( \{k_1, \ldots, k_\ell, \ldots, k_n\} \), and can assume only the values \( k_\ell = 0, 1 \).

Now we can define our generalization of quantum Fourier transform.

**Definition 13.** Let \( |\ell\rangle \) be an element of the computational basis of a \( n \)-qubit space representing the integer \( \ell \) by the following rule:

\[
\ell = \sum_{r=1}^{n} \ell_r 2^{n-r}; \quad \ell_r = 0, 1. \quad (38)
\]

Then the Clifford quantum Fourier transform of \( |j\rangle \) is defined by the map

\[
|j\rangle \mapsto \frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi ij \frac{k}{2^n} I + i \theta \Gamma_k} |k\rangle, \quad (39)
\]

where \( I \) is the identity matrix, \( j \) and \( k \) are integers and \( \theta \in \mathbb{R} \).

**Remark 9.** For the sake of notation, we will sometimes omit the identity matrix \( I \) in the Definition 13 when the context is clear.
Remark 10. Note that the Clifford quantum Fourier transform (39) can be rewritten in the form

\[ F_N = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} e^{2\pi i j k} I + i\theta \Gamma_k |k\rangle \langle k|, \quad (40) \]

where \( N = 2^n \).

We can expand the definition of the Clifford quantum Fourier transform (39) considering the binary expansion of \( j \) and \( k \) by (38) and the definition of \( \Gamma_k \) given by (37). We obtain the following expression

\[
|j\rangle \mapsto \frac{1}{2^{n/2}} \sum_{k_1=0}^{2^{n/2}} \cdots \sum_{k_n=0}^{2^{n/2}} e^{2\pi i j (\sum_{l=1}^{n} k_l 2^{-l})} e^{i\theta([\alpha_{k_1}^0 \sigma_x + \alpha_{k_1}^1 \sigma_y + \alpha_{k_1}^2 \sigma_z] \otimes \cdots \otimes I \cdots \otimes [\alpha_{k_n}^0 \sigma_x + \alpha_{k_n}^1 \sigma_y + \alpha_{k_n}^2 \sigma_z])} |k_1 \cdots k_n\rangle
\]

where \(|k_1 \cdots k_n\rangle = |k_1\rangle \otimes |k_2\rangle \otimes \cdots \otimes |k_n\rangle\). Since all the parcels of \( \Gamma_k \)'s do commute among them and with \( \frac{2\pi i j}{2^n} I \), this expression can be factored into

\[
\frac{1}{2^{n/2}} \sum_{k_1=0}^{2^{n/2}} \cdots \sum_{k_n=0}^{2^{n/2}} e^{2\pi i j (\sum_{l=1}^{n} k_l 2^{-l})} e^{i\theta([\alpha_{k_1}^0 \sigma_x + \alpha_{k_1}^1 \sigma_y + \alpha_{k_1}^2 \sigma_z] \otimes \cdots \otimes I \cdots \otimes [\alpha_{k_n}^0 \sigma_x + \alpha_{k_n}^1 \sigma_y + \alpha_{k_n}^2 \sigma_z])} |k_1 \cdots k_n\rangle = \sum_{k_1=0}^{2^{n/2}} \cdots \sum_{k_n=0}^{2^{n/2}} e^{2\pi i j (\sum_{l=1}^{n} k_l 2^{-l})} e^{i\theta([\alpha_{k_1}^0 \sigma_x + \alpha_{k_1}^1 \sigma_y + \alpha_{k_1}^2 \sigma_z] \otimes \cdots \otimes I \cdots \otimes [\alpha_{k_n}^0 \sigma_x + \alpha_{k_n}^1 \sigma_y + \alpha_{k_n}^2 \sigma_z])} |k_1 \cdots k_n\rangle
\]

Then the Clifford quantum Fourier transform of \(|j\rangle\) is

\[
|j\rangle \mapsto \frac{1}{2^{n/2}} \left[ e^{i\theta([\alpha_{j_1}^0 \sigma_x + \alpha_{j_1}^1 \sigma_y + \alpha_{j_1}^2 \sigma_z] \otimes \cdots \otimes I \cdots \otimes [\alpha_{j_n}^0 \sigma_x + \alpha_{j_n}^1 \sigma_y + \alpha_{j_n}^2 \sigma_z])} |0\rangle + e^{2\pi i j 2^{-n}} e^{i\theta([\alpha_{j_1}^0 \sigma_x + \alpha_{j_1}^1 \sigma_y + \alpha_{j_1}^2 \sigma_z] \otimes \cdots \otimes I \cdots \otimes [\alpha_{j_n}^0 \sigma_x + \alpha_{j_n}^1 \sigma_y + \alpha_{j_n}^2 \sigma_z])} |1\rangle \right].
\]
The unitary rotation operator can be expressed as \[25\]

\[
\text{Proof. I}
\]

Lemma 2. Let \(R_{\vec{n}}(\theta)\) be a unitary rotation operator and \(|k\rangle\) an element of the computational basis of the space of \(n\)-qubits. Then \[
\sum_{k=0}^{1} R_{\vec{n}}(\theta)|k\rangle \langle k| R^\dagger_{\vec{n}}(\theta) = I.
\]

Proof. The unitary rotation operator can be expressed as \[25\]

\[
R_{\vec{n}}(\theta) = \exp \left( -i\theta \vec{n} \cdot \vec{\sigma} / 2 \right) = \cos \left( \theta / 2 \right) I - i \sin \left( \theta / 2 \right) (n_x \sigma_x + n_y \sigma_y + n_z \sigma_z),
\]

(43)

where \((n_x, n_y, n_z)\) is a real unit vector. Therefore we, by a straightforward calculation, check the result of the lemma:

\[
\sum_{k=0}^{1} R_{\vec{n}}(\theta)|k\rangle \langle k| R^\dagger_{\vec{n}}(\theta) = I.
\]

(44)

Theorem 14. The Clifford quantum Fourier transform \(F_N = \frac{1}{\sqrt{N}} \sum_{j,k=0}^{N-1} e^{-2\pi i jk / N} e^{i\theta_j k} |k\rangle \langle j|\) is unitary.
Proof. We have that $F^+_N$ is given by

$$F^+_N = \frac{1}{\sqrt{N}} \sum_{j, k=0}^{N-1} e^{-2\pi i j' k' / N} |j'\rangle \langle k'| e^{-i\theta \Gamma k'}. \quad (45)$$

Consequently,

$$F_N F^+_N = \frac{1}{N} \sum_{j, k=0}^{N-1} \sum_{j', k'=0}^{N-1} e^{2\pi i (j-j' / N)} e^{i\theta \Gamma k} |j\rangle \langle j'| e^{-i\theta \Gamma k'}$$

$$= \frac{1}{N} \sum_{j, k=0}^{N-1} \sum_{j', k'=0}^{N-1} e^{2\pi i (j-j' / N)} |j\rangle \langle j'| e^{i\theta \Gamma k} e^{-i\theta \Gamma k'}$$

$$= \frac{1}{N} \sum_{k, k'} N\delta_{kk'} e^{i\theta \Gamma k} |k\rangle \langle k'| e^{-i\theta \Gamma k'}$$

$$= \sum_k e^{i\theta \Gamma k} |k\rangle \langle k| e^{-i\theta \Gamma k}, \quad (46)$$

where we used $\sum_j e^{2\pi i (k-k') / N} = N\delta_{kk'}$. Therefore

$$F_N F^+_N = \sum_{k_1, \ldots, k_n=0}^{1} e^{i\theta (\alpha_{x_1} \sigma_x + \alpha_{y_1} \sigma_y + \alpha_{z_1} \sigma_z) \otimes \cdots \otimes (\alpha_{x_n} \sigma_x + \alpha_{y_n} \sigma_y + \alpha_{z_n} \sigma_z)}$$

$$\times |k_1 \cdots k_n\rangle \langle k_1 \cdots k_n|$$

$$\times \sum_{k_1, \ldots, k_n=0}^{1} e^{i\theta (\alpha_{x_1} \sigma_x + \alpha_{y_1} \sigma_y + \alpha_{z_1} \sigma_z) \otimes \cdots \otimes (\alpha_{x_n} \sigma_x + \alpha_{y_n} \sigma_y + \alpha_{z_n} \sigma_z)}$$

$$= \left[ \sum_{k_1=0}^{1} e^{i\theta (\alpha_{x_1} \sigma_x + \alpha_{y_1} \sigma_y + \alpha_{z_1} \sigma_z)} |k_1\rangle \langle k_1| e^{-i\theta (\alpha_{x_1} \sigma_x + \alpha_{y_1} \sigma_y + \alpha_{z_1} \sigma_z)} \right]$$

$$\otimes \cdots \otimes$$

$$\left[ \sum_{k_n=0}^{1} e^{i\theta (\alpha_{x_n} \sigma_x + \alpha_{y_n} \sigma_y + \alpha_{z_n} \sigma_z)} |k_n\rangle \langle k_n| e^{-i\theta (\alpha_{x_n} \sigma_x + \alpha_{y_n} \sigma_y + \alpha_{z_n} \sigma_z)} \right].$$

Using the result of the Lemma 2, we get

$$F_N F^+_N = (|0\rangle \langle 0| + |1\rangle \langle 1|) \otimes \cdots \otimes (|0\rangle \langle 0| + |1\rangle \langle 1|)$$

$$= I^\otimes n. \quad (47)$$

Analogously, it is easy to verify that $F^+_N F_N = I^\otimes n$. With these results, we proved the theorem. \qed

We can see Clifford quantum Fourier transform can be implemented via group theory. Consider the group $spin_+(3, 0)$ (see Appendix A for the definition of this group) and its respectively Lie algebra $Lie[spin_+(3, 0)]$. We know from Lie group and Lie algebra theories that if we have a tensor product of many groups, in our case this group is $spin_+(3, 0)$, we would have

$$\bigotimes_{i=1}^{n} spin_+(3, 0) \simeq spin_+(3, 0) \otimes \cdots \otimes spin_+(3, 0). \quad (48)$$
Then, the Lie algebra of this tensor product of groups (48) is the direct sum of the Lie algebras of the group spin\(_+\)\((3,0)\),

\[
\bigoplus_{i=1}^{n} \text{Lie}[\text{spin}_+ (3,0)] \simeq \text{Lie}[\text{spin}_+ (3,0)] \oplus \cdots \oplus \text{Lie}[\text{spin}_+ (3,0)]. \quad (49)
\]

The next theorem relates the group (48), the Lie algebra (49) and their representations:

**Theorem 15.** Let \( \rho \) and \( \rho' \) be two representations of, respectively, \( \left( [\Lambda^2 \mathbb{R}^{p,q}]^{\oplus n}, [\ , \ ] \right) \) and the universal enveloping algebra of the Lie algebra of the group spin\(_+\)(p,q), given by

\[
U(\text{Lie}[\text{spin}_+ (p,q)]) = \frac{T[\text{Lie}[\text{spin}_+ (p,q)]]}{I_d},
\]

where \( T \) is the tensor algebra and \( I_d \) is the two-sided ideal generated by elements of the form \( x \otimes y - y \otimes x - [x,y] \). Then

\[
\rho'(\text{Lie}[\text{spin}_+ (p,q)]) \otimes I \otimes \cdots \otimes I + \cdots + I \otimes \cdots \otimes I \otimes \rho'(\text{Lie}[\text{spin}_+ (p,q)]) \\
\simeq \rho \left( [\Lambda^2 \mathbb{R}^{p,q}]^{\oplus n}, [\ , \ ] \right).
\]

**Proof.** Initially, we prove that \( \left( [\Lambda^2 \mathbb{R}^{p,q}]^{\oplus n}, [\ , \ ] \right) \) is the direct sum \( \bigoplus_{i=1}^{n} \text{Lie}[\text{spin}_+ (p,q)] \). First consider \( B = (B_1, B_2, \ldots, B_n) \) and \( B' = (B'_1, B'_2, \ldots, B'_n) \) bivectors in \( [\Lambda^2 \mathbb{R}^{p,q}]^{\oplus n} \), where \( B_i \) and \( B'_i \), \( 1 \leq i \leq n \), correspond to the elements of the \( i \)th parcel of the direct sum space. We can define the Lie product in \( [\Lambda^2 \mathbb{R}^{p,q}]^{\oplus n} \) as \( [B, B'] = ([B_1, B'_1], \ldots, [B_n, B'_n]) \in [\Lambda^2 \mathbb{R}^{p,q}]^{\oplus n} \), with \( [B_i, B_j] \equiv B_i B_j - B_j B_i \). Then,

\[
BB' = (B_1 B'_1)_0 + (B_1 B'_1)_1 + \cdots + (B_1 B'_1)_4, \ldots, (B_n B'_n)_0 + (B'_n B_n)_2 + (B'_n B_n)_4
\]

\[

eq [\mathbb{R} \oplus \bigwedge^2 \mathbb{R}^{p,q} \oplus \bigwedge^4 \mathbb{R}^{p,q}]^{\oplus n}. \quad (51)
\]

We have that

\[
(BB')^\sim = ((B_1 B'_1)^\sim, \ldots, (B_n B'_n)^\sim)
\]

\[
= ((B'_1 B_1)_0 - (B'_1 B_1)_1 + (B'_1 B_1)_4, \ldots, (B'_n B_n)_0 - (B'_n B_n)_2 + (B'_n B_n)_4).
\]

On the other hand,

\[
(BB')^\sim = ((B_1 B'_1)^\sim, \ldots, (B_1 B'_1)^\sim)
\]

\[
= (\tilde{B}_1 B'_1, \ldots, \tilde{B}_n B'_n)
\]

\[
= (B_1 B_1, \ldots, B_n B_n). \quad (52)
\]

Consequently,

\[
[B, B'] = (2(B_1 B'_1)_2, \ldots, 2(B_n B'_n)_2). \quad (53)
\]
With this result, we have shown that
\[
\left( \bigwedge^2 \mathbb{R}^{n,q}, [ , ] \right) \oplus \cdots \oplus \left( \bigwedge^2 \mathbb{R}^{n,q}, [ , ] \right) \cong \bigoplus_{i=1}^{n} \text{Lie} [\text{spin}_+(p,q)]. \tag{54}
\]

Now we define the map \( \psi \) given by:
\[
\psi(\rho(B_1, \ldots, B_n)) = \rho'(B_1) \otimes I \otimes \cdots \otimes I + \cdots + I \otimes \cdots \otimes I \otimes \rho'(B_n). \tag{55}
\]
where we take account the isomorphism (55) to use the abuse of notation \( b_j \equiv B_j, c_j \equiv C_j \), with \( b_j, c_j \in \text{spin}_+(p,q) \). Thus,
\[
\begin{align*}
\psi([\rho(B_1, \ldots, B_n), \rho(C_1, \ldots, C_n)]) & = \psi(\rho([B_1, C_1], \ldots, [B_n, C_n])) \\
& = \rho'([B_1, C_1]) \otimes I \otimes \cdots \otimes I + \cdots + I \otimes \cdots \otimes I \otimes \rho'([B_n, C_n]) \\
& = \rho'(B_1 C_1 - C_1 B_1) \otimes I \otimes \cdots \otimes I + \cdots + I \otimes \cdots \otimes I \otimes \rho'(B_n C_n - C_n B_n)
\end{align*}
\]
since we considered universal enveloping algebra. Therefore,
\[
\begin{align*}
\psi([\rho(B_1, \ldots, B_n), & \rho(C_1, \ldots, C_n)]) \\
& = \rho'(B_1) \rho'(C_1) \otimes I \otimes \cdots \otimes I - \rho'(C_1) \rho'(B_1) \otimes I \otimes \cdots \otimes I \\
& \quad + \cdots + I \otimes \cdots \otimes I \otimes \rho'(B_n) \rho'(C_n) - I \otimes \cdots \otimes I \otimes \rho'(C_n) \rho'(B_n) \\
& = [\rho'(B_1) \otimes I \otimes \cdots \otimes I + \cdots + I \otimes \cdots \otimes I \otimes \rho'(B_n)] \\
& \quad \times [\rho'(C_1) \otimes I \otimes \cdots \otimes I + \cdots + I \otimes \cdots \otimes I \otimes \rho'(C_n)] \\
& \quad - [\rho'(C_1) \otimes I \otimes \cdots \otimes I + \cdots + I \otimes \cdots \otimes I \otimes \rho'(C_n)] \\
& \quad \times [\rho'(B_1) \otimes I \otimes \cdots \otimes I + \cdots + I \otimes \cdots \otimes I \otimes \rho'(B_n)] \\
& = [\psi(\rho(B_1, \ldots, B_n)), \psi(\rho(C_1, \ldots, C_n))].
\end{align*}
\]
Then we have a bijective homomorphism and the proof is finished. \( \square \)

This result is general and we can particularize to \( \mathbb{R}^{3,0} \). This explains the choice of the \( \Gamma_k \) elements (37) in the definition of the Clifford quantum Fourier transform (39) and why it can be factored in \( n \) factors: the \( i\Gamma_k \)'s are elements of a representation of the direct sum of the algebras \([\bigwedge \mathbb{R}^{3,0}, [ , ] ] \cong \text{Lie} [\text{spin}_+(3,0)]\), which is equivalent to the \( n \) direct products of of \( \text{spin}_+(3,0) \).

A typical example of an element of a representation of the latter group is the exponential \( e^{i\theta \Gamma_k} \), added in the standard quantum Fourier transform.

The next theorem gives us the distance between our Clifford quantum Fourier transform and the usual quantum Fourier transform.

**Theorem 16.** Let \( F_N \) be Clifford quantum Fourier transform and \( \mathcal{F}_N \) be the usual quantum Fourier transform. Then
\[
\| F_N - \mathcal{F}_N \| \leq 2 \frac{\log N}{N} \theta \log N \sqrt{2e^\theta \log N \sqrt{2}}, \tag{56}
\]
where \( \| A \| = \sqrt{\sum_{i,j=1}^{n} |a_{ij}|^2} \) [18].
Proof. We have that

\[
\|F_N - \mathcal{F}_N\| = \left\| \frac{1}{\sqrt{N}} \sum_{j,k=0}^{N-1} e^{\frac{2\pi i j k}{N} + i \theta \Gamma_k} |k\rangle \langle j| - \frac{1}{\sqrt{N}} \sum_{j,k=0}^{N-1} e^{\frac{2\pi i j k}{N}} |k\rangle \langle j| \right\|
\]

\[
= \left\| \frac{1}{\sqrt{N}} \sum_{j,k=0}^{N-1} \left( e^{\frac{2\pi i j k}{N} + i \theta \Gamma_k} - e^{\frac{2\pi i j k}{N}} \right) |k\rangle \langle j| \right\|
\]

\[
\leq \frac{1}{\sqrt{N}} \sum_{j,k=0}^{N-1} \left\| \left( e^{\frac{2\pi i j k}{N} + i \theta \Gamma_k} - e^{\frac{2\pi i j k}{N}} \right) |k\rangle \langle j| \right\|
\]

\[
= \frac{1}{\sqrt{N}} \sum_{j,k=0}^{N-1} \left\| e^{\frac{2\pi i j k}{N} + i \theta \Gamma_k} - e^{\frac{2\pi i j k}{N}} \right\|_{I \otimes \cdots \otimes I} \cdot \left\| |k\rangle \langle j| \right\|
\]

\[
\leq \frac{1}{\sqrt{N}} \sum_{j,k=0}^{N-1} \left\| e^{\frac{2\pi i j k}{N} + i \theta \Gamma_k} - e^{\frac{2\pi i j k}{N}} \right\|_{I \otimes \cdots \otimes I}.
\] (57)

Using \(\|e^X + Y - e^X\| \leq \|Y\|_2\|e^Y\|\) [19],

\[
\|F_N - \mathcal{F}_N\| \leq \frac{1}{\sqrt{N}} \sum_{j,k=0}^{N-1} \|i \theta \Gamma_k\|_2 \|e^{\frac{2\pi i j k}{N} / I \otimes \cdots \otimes I} \|_2 \|e^{\frac{2\pi i j k}{N}} \|_2.
\] (58)

Since \(\|X \otimes Y\| = \|X\| \cdot \|Y\|\), then we have

\[
\|F_N - \mathcal{F}_N\| \leq \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \|i \theta \left( (\alpha_{x_1}^{k_1} \sigma_x + \alpha_{y_1}^{k_1} \sigma_y + \alpha_{z_1}^{k_1} \sigma_z) \right. \left. \otimes I \otimes \cdots \otimes I \langle \alpha_{x_1}^{k_1} \sigma_x + \alpha_{y_1}^{k_1} \sigma_y + \alpha_{z_1}^{k_1} \sigma_z) \right) \|_2 \|e^{\frac{2\pi i j k_1}{N} 2^{-l} / I \otimes \cdots \otimes I} \|_2 \times \|e^{\frac{2\pi i j k_{l+1}}{N} 2^{-l} / I \otimes \cdots \otimes I} \|_2 \times \cdots \times \|e^{\frac{2\pi i j k_n}{N} 2^{-l} / I \otimes \cdots \otimes I} \|_2
\]

\[
\leq \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} 2^n \theta n \sqrt{2} e^{\theta n \sqrt{2}}
\]

\[
\leq 2^{\frac{3n}{2}} \theta n \sqrt{2} e^{\theta n \sqrt{2}}
\]

\[
= 2^{\frac{3n}{2}} \theta \log N \sqrt{2} e^{\theta \log N \sqrt{2}}.
\] (59)

\[\square\]

Note that the distance between the standard and our quantum Fourier transform is upper bounded by the number of qubits used in their implementation. However, as its bound depends on the exponential of the number of qubits, our analysis suggests this distance can be considerably large.

To finish this section, it is worth noting that although our aim here is to use the parameters of the Clifford quantum Fourier transform in the models of quantum neural networks developed in Sect. 2, it has potential to be used in other applications such as quantum information processing and
development of novel quantum algorithms. Nonetheless, our Clifford generalization of quantum Fourier transform has itself a mathematical importance, especially because of its relationship with Lie groups and algebras. These and other aspects of the Clifford quantum Fourier transform may be investigated in future works.

5. Conclusions

In this paper, we present several models of quantum neural networks using suitable representations of the Clifford algebras $Cl(2n)$ and tensor product of $Cl(3,0)^{\otimes n}$ in order to take geometric and topological advantages allowed by these algebras. We showed, within a rigorous mathematical framework, that these algebraic structures allow general unitary learning algorithms and arbitrary activation functions implemented by a quantum-classical hybrid scheme. The basic idea is that unitary operators can be constructed from representations of Clifford algebras and then the elements of these algebras can be associated with geometric objects. In this sense, encoded multivectors make it possible controlling of subspaces without giving up information about their orientations [24], which may be helpful in multidimensional data analysis. Therefore, our models are able to capture geometric information contained in the data and generate patterns involving quantum entanglement. In this context, Hamiltonian simulation techniques allow error analysis of possible approximations of the unitary operations constructed by exponentials of general Clifford elements. Consequently, our formulations allow machine learning of quantum systems, since we can build arbitrary Hamiltonians based on representations of Clifford algebras. Besides, we also proposed an algebraic generalization of the quantum Fourier transform that contains additional parameters that enable its use as quantum neural network models. Important properties such as unitarity, factorization, and upper bound for the distance between ours and the standard quantum Fourier transform were derived. It is important to point out that applications of this generalized transform can go beyond quantum machine learning. For instance, we can note our quantum Fourier transform may be useful in quantum information processing since one can use its $3n$ arbitrary additional parameters to encode, process and transmit information. We can also point out that this generalized approach makes it possible to build specific models based on spinors and groups related to Clifford algebras such as the classical Clifford neurons [9]. An important aspect to be highlighted is that since $spin(n)$ is the double cover of $SO(n)$ and $Lie(spin(n)) \simeq so(n)$ we can build representations of orthogonal neural networks using Clifford algebras in a systematic way. These networks may evade explosive or evanescent gradients [22]. Another interesting feature of our models is that, in some of them, we can infer the derivatives related to weight parameters with measures of suitable operators in quantum computers. Finally, we believe that this proposal may be useful in the near-term quantum computing and as perspectives, we intend to perform numerical
simulations, compare performance with other models, and implement our models in quantum computers.

Appendix

A Basic Elements of Clifford Algebras

In this appendix, we review some basic concepts about Clifford algebras [24, 32].

Given a vector space $V$, the Clifford algebra can be defined as quotient $\text{Cl}(V, Q) = \frac{T(V)}{I_0}$, where $I_0$ is a two sided ideal generated by elements $v \otimes v - Q(v)1$, for all $v \in V$; $Q$ is the quadratic form and $T(V)$ is the tensor algebra. Alternatively, let $V$ be a space vector over $\mathbb{R}$ equipped with a symmetric bilinear form $g$, $A$ an associative algebra with unit $1_A$, and $\gamma$ a linear application $\gamma: V \to A$. The pair $(A, \gamma)$ is a Clifford algebra for the quadratic space $(V, g)$ if $A$ is generated as an algebra by $\{\gamma(v); v \in V\}$, $\{a1_A; a \in \mathbb{R}\}$ and satisfies

$$\gamma(v)\gamma(u) + \gamma(u)\gamma(v) = 2g(v, u)1_A,$$

for all $v, u \in V$. Let $V$ be a vector space $\mathbb{R}^n$ and $g$ a symmetric bilinear form in $\mathbb{R}^n$ of signature $(p, q)$ with $p + q = n$. We will denote by $\text{Cl}(p, q) \equiv \text{Cl}_{p,q} \equiv \text{Cl}(\mathbb{R}^{p,q})$ the Clifford algebra associated with the quadratic space $\mathbb{R}^{p,q}$. In addition, we denote $\text{Cl}(2n, 0) \equiv \text{Cl}(2n)$. The even subalgebra is defined by:

$$\text{Cl}^{+}_{p,q} = \{\Gamma \in \text{Cl}_{p,q}; \Gamma = \hat{\Gamma}\},$$

where $\hat{\cdot}$ denotes graded involution, which keeps the sign of the elements belonging to even subspaces. Note that the involution of a $k$-vector $\Gamma^{[k]}$ is given by $\hat{\Gamma}^{[k]} = (-1)^{k(k-1)/2}\Gamma^{[k]}$. The groups $\text{spin}(p, q)$ and $\text{spin}^+(p, q)$ are given by:

$$\text{spin}(p, q) = \{a \in \text{Cl}^+_{p,q}; N(a) = \pm 1\}$$

and

$$\text{spin}^+(p, q) = \{a \in \text{Cl}^+_{p,q}; N(a) = 1\},$$

respectively, where $N(a) = |a|^2 = \langle \bar{a}a \rangle_0$ is related to the norm of elements of Clifford algebra, and $\bar{\cdot}$ represents the reversion operator defined by $\bar{\Gamma}^{[k]} = (-1)^{k(k+1)/2}\Gamma^{[k]}$; $\langle \cdot \rangle_k : \text{Cl}_{p,q} \to \bigwedge_k(\mathbb{R}^{p,q})$, with $\bigwedge_k(\mathbb{R}^{p,q})$ denoting the exterior algebra of vector space $\mathbb{R}^{p,q}$.

The Lie Algebra of $\text{spin}^+(p, q)$ is the space of bivectors $\bigwedge^2 \mathbb{R}^{p,q}$. Let $B_1$ and $B_2$ be two bivectors. Then the commutator

$$[B_1, B_2] \in \bigwedge^2 \mathbb{R}^{p,q}$$

is a bivector.
B Swap Test

The swap test consists of the decomposition [34]

\[(\mathbb{C}^d)^{\otimes 2} = \text{Sym}^2(\mathbb{C}^d) \oplus \Lambda^2(\mathbb{C}^d),\]  

(66)

where \(\text{Sym}^2(\mathbb{C}^d)\) and \(\Lambda^2(\mathbb{C}^d)\) are symmetric and antisymmetric spaces, respectively. Consider the system in the state \(|0, \phi, \psi\rangle\). The Hadamard gate transforms this state in

\[\frac{1}{\sqrt{2}}(|0, \phi, \psi\rangle + |1, \phi, \psi\rangle).\]  

(67)

Then, the controlled swap gate produces

\[\frac{1}{\sqrt{2}}(|0, \psi, \phi\rangle + |1, \phi, \psi\rangle).\]  

(68)

After a second application of Hadamard gate, we obtain

\[\frac{1}{2}|0\rangle(|\psi, \phi\rangle + |\phi, \psi\rangle) + \frac{1}{2}|1\rangle(|\psi, \phi\rangle - |\phi, \psi\rangle).\]  

(69)

If we perform a measurement on the first qubit, we would get

\[\Pr(\text{outcome} = 0) = \frac{1}{2} \left(1 + |\langle \psi | \phi \rangle|^2\right).\]  

(70)

By repeating \(N\) times this procedure, we obtain

\[|\langle \psi | \phi \rangle| = \left(1 - \frac{2#\{\text{outcome} = 0\}}{N}\right)^{1/2}.\]  

(71)

This equation allows the calculation of quantum fidelity in a quantum computer through measurements on the first qubit.

C A Simple Example of Decomposition of Quantum Gates

Unitary operators can be obtained through exponentials of anti-Hermitian operators. We will show how a simple circuit can be built from exponentials of elements of representations of Clifford algebras following the prescription obtained in the reference [25]. Thus we will build a circuit related to the unitary transformation

\[U(\theta_1, \theta_2) = \exp[i(\theta_1(\sigma_x \otimes \sigma_y) + \theta_2(\sigma_y \otimes \sigma_x))]\]

\[= U(\theta_1)U(\theta_2),\]

where

\[U(\theta_1) = \begin{pmatrix}
\cos(\theta_1) & 0 & 0 & \sin(\theta_1) \\
0 & \cos(\theta_1) & \sin(\theta_1) & 0 \\
0 & -\sin(\theta_1) & \cos(\theta_1) & 0 \\
-\sin(\theta_1) & 0 & 0 & \cos(\theta_1)
\end{pmatrix},\]

and
\[ U(\theta_2) = \begin{pmatrix}
\cos(\theta_2) & 0 & 0 & \sin(\theta_2) \\
0 & \cos(\theta_2) & \sin(\theta_2) & 0 \\
0 & -\sin(\theta_2) & \cos(\theta_2) & 0 \\
-\sin(\theta_2) & 0 & 0 & \cos(\theta_2)
\end{pmatrix}. \]

\[ U(\theta_1) \text{ can be expressed as a product of two-level unitary gates} \]
\[ U(\theta_1) = U_1(\theta_1)U_2(\theta_1)U_3(\theta_1), \quad (72) \]

where
\[ U_1(\theta_1) = \begin{pmatrix}
\cos(\theta_1) & 0 & 0 & -\sin(\theta_1) \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
-\sin(\theta_1) & 0 & 0 & \cos(\theta_1)
\end{pmatrix}, \]
\[ U_2(\theta_2) = \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & \cos(\theta_2) & \sin(\theta_2) & 0 \\
0 & \sin(\theta_2) & -\cos(\theta_2) & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}, \]

and
\[ U_3(\theta_2) = \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & 0 & -1
\end{pmatrix}. \]

Analogously, \( U(\theta_2) \) can be expressed as a product of two-level unitary gates
\[ U(\theta_2) = U_1(\theta_2)U_2(\theta_2)U_3(\theta_2), \quad (73) \]

where
\[ U_1(\theta_2) = \begin{pmatrix}
\cos(\theta_1) & 0 & 0 & -\sin(\theta_1) \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
-\sin(\theta_1) & 0 & 0 & \cos(\theta_1)
\end{pmatrix}, \]
\[ U_2(\theta_1) = \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & \cos(\theta_1) & -\sin(\theta_1) & 0 \\
0 & -\sin(\theta_1) & -\cos(\theta) & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}, \]

and
\[ U_3(\theta_1) = \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & 0 & -1
\end{pmatrix}, \]

so that
\[ U(\theta_1, \theta_2) = U(\theta_1)U(\theta_2) = U_1(\theta_1)U_2(\theta_1)U_3(\theta_1)U_1(\theta_2)U_2(\theta_2)U_3(\theta_2). \tag{74} \]

The circuit is illustrated in the figure below.

where
\[
\tilde{U}_1(\theta_1) = \begin{pmatrix} \cos(\theta_1) & -\sin(\theta_1) \\ -\sin(\theta_1) & \cos(\theta_1) \end{pmatrix}, \quad \tilde{U}_2(\theta_1) = \begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \sin(\theta_1) & -\cos(\theta_1) \end{pmatrix},
\]

and
\[
\tilde{U}_1(\theta_2) = \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) \\ -\sin(\theta_2) & \cos(\theta_2) \end{pmatrix}, \quad \tilde{U}_2(\theta_2) = \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) \\ -\sin(\theta_2) & \cos(\theta_2) \end{pmatrix},
\]

Notice that
\[
U(\theta_1)U(\theta_2)|00\rangle = [\cos(\theta_1)\cos(\theta_2) - \sin(\theta_1)\sin(\theta_2)]|00\rangle
+ [\sin(\theta_1)\cos(\theta_2) - \sin(\theta_2)\cos(\theta_1)]|11\rangle, \tag{75}
\]

which is generally an entangled state. We will build states with the following notation:
\[
|x; \theta\rangle = U(\theta)U(x)|00\rangle
= [\cos(\theta)\cos(x) - \sin(\theta)\sin(x)]|00\rangle + [\sin(\theta)\cos(x) - \sin(x)\cos(\theta)]|11\rangle,
\]

where \( \theta_1 = \theta \) and \( \theta_2 = x \) and
\[
|y\rangle = U(y)|00\rangle = \cos(y)|00\rangle - \sin(y)|11\rangle.
\]

Data availability statement  No data is used to support this article.

Declarations

Conflict of interest  The authors declare that they have no conflict of interest or competing interests.

Publisher’s Note  Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.
References

[1] Altaisky, M.V.: Quantum neural network, Technical Report, Joint Institute for Nuclear Research, Russia (2001). arXiv: quant-ph/0107012v2
[2] Alves, R., Lavor, C.: Clifford Algebra applied to Grover’s Algorithm. Adv. Appl. Clifford Alg. 20, 477–488 (2010)
[3] Bayro-Corrochano, E.: Applications of Geometric Algebra Vol. I. Computer Vision, Graphics and Neurocomputing, Chap 13, pp. 471–478. Springer, Berlin (2019)
[4] Bayro-Corrochano, E., Gamboa-Soliz, S., Altamirano-Escobedo, G., Lechuga-Gutierrez, L., Lisarraga-Rodriguez, J.: Quaternionic spiking and quaternionic quantum neural networks: theory and applications. Int. J. Neural Syst. 31, 2 (2021)
[5] Beer, K., Bondarenko, D., Farrelly, T., Osborne, J., Salzmann, R., Scheiermann, D., Wolf, R.: Training deep quantum neural networks. Nat. Commun. 11(1), 808 (2020)
[6] Benedetti, M., Lloyd, E., Sack, S., Fiorentini, M.: Parametrized quantum circuits as machine learning models. Quant. Sci. Technol. 4(4), 043001 (2019)
[7] Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S.: Quantum machine learning. Nature 549, 195–202 (2017)
[8] Bottaci, L.: Artificial neural networks applied to outcome prediction for colorectal cancer patients in separate institutions. Lancet 350(9076), 469–72 (1997)
[9] Buchholz, S.: A Theory of Neural Computation with Clifford Algebras. PhD Thesis, Kiel (2005)
[10] Buchholz, S., Sommer, G.: On Clifford neurons and multi-layer perceptrons. Neural Netw. 21, 925–935 (2008)
[11] Cai, X.-D., et al.: Entanglement-based machine learning on a quantum computer. Phys. Rev. Lett. 1144, 110504 (2015)
[12] Childs, A.M., Maslov, D., Nam, Y., Ross, N.J., Su, Y.: Toward the first quantum simulation with quantum speedup. Proc. Natl. Acad. Sci. 115(38), 9456–9461 (2017)
[13] Ciresan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: IEEE Conference on Computer vision and Pattern Recognition, pp. 3642–3649 (2012)
[14] da Silva, A.J., Ludermir, T.B., Oliveira, W.R.: Quantum perceptron over a field and neural network architecture in quantum computer. Neural Netw. 76, 55–64 (2016)
[15] De Bie, H.: Clifford algebras, Fourier transforms, and quantum mechanics. Math. Methods Appl. Sci. 35(18), 2198–2228 (2012)
[16] Ganesan, N.: Application of neural networks in diagnosing cancer disease using demographic data. Int. Comput. Appl. 1(26), 81–97 (2010)
[17] Harrow, A.W., Hassidim, A., Lloyd, S.: Quantum algorithm for linear systems of equations. Phys. Rev. Lett. 103, 150502 (2009)
[18] Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge (1990)
[19] Horn, R.A., Johnson, C.R.: Topics in Matrix Analysis. Cambridge University Press, Cambridge (1991)
[20] Jozsa, R., Miyake, A.: Matchgates and classical simulation of quantum circuits. Proc. R. Soc. A 464, 3089–3106 (2008)
[21] Kak, S.C.: Quantum neural computing. Adv. Imaging Electron Phys. 94, 259–313 (1995)
[22] Kerenidis, I., Landman, J., Mathur, N.: Classical and Quantum Algorithms for orthogonal neural networks (2021). arXiv:2106.07198v1 [quant-ph]
[23] Lloyd, S., Mohseni, M., Rebentrost, P.: Quantum principal component analysis. Nat. Phys. 10, 631–633 (2014)
[24] Lounesto, P.: Clifford Algebras and Spinors, 2nd edn. Cambridge University Press, Cambridge (2001)
[25] Nielsen, M., Chuang, I.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)
[26] Rosenblatt, F.: The Perceptron—a perceiving and recognizing automaton, Report 85-460-1. Cornell Aeronautical Laboratory (1957)
[27] Schuld, M.: Petruccione. Machine Learning with Quantum Computers. Springer, Switzerland (2021)
[28] Shao, Chapeng: A quantum model of feed-forward neural networks with unitary learning algorithms. Quant. Inf. Process. 19, 102 (2020)
[29] Song, Hai-Jing., Zhou, D.L.: Group theory on quantum Boltzmann machine. Phys. Lett. A 399, 127298 (2021)
[30] Teguri, T., Iosokawa, T., Matsui, N., Nishimura H., Kamiura, N.: Time series prediction by quaternionic qubit neural network. In: International Joint Conference on Neural Networks (IJCNN), pp. 1–6 (2020)
[31] Trindade, M.A.S., Pinto, E., Vianna, J.D.M.: Adv. Appl. Clifford Alg. 26, 771–792 (2016)
[32] Vaz, J., Jr., Rocha, R., Jr.: An Introduction to Clifford Algebras and Spinors. Oxford University Press, New York (2016)
[33] Vlasov, A.Y.: Clifford algebras and universal quantum gates. Phys. Rev. A 63, 054302 (2001)
[34] Walter, M.: Symmetry and Quantum Information. Lecture Notes, Spring (2018)
[35] Wehner, S., Winter, A.: Higher entropic uncertainty relations for anti-commuting observables. J. Math. Phys. 49, 062105 (2008)
[36] Zhang, G., Rong, H., Neri, F., Prez-Jimnez, M.J.: An optimization spiking neural system for approximately solving combinatorial optimizations problems. Int. J. Neural Syst. 24(05), 1440006 (2014)
Marco A. S. Trindade  
Colegiado de Física, Departamento de Ciências Exatas e da Terra  
Universidade do Estado da Bahia  
41150-000 Salvador, BA  
Brazil  
e-mail: matrindade@uneb.br

Vinícius N. A. Lula-Rocha  
Atos, Latin American Quantum Computing Center  
41650-010 Salvador, BA  
Brazil  
e-mail: viniciusnonato@gmail.com

S. Floquet  
Colegiado de Engenharia Civil  
Universidade Federal do Vale São Francisco  
48,902-300 Juazeiro, BA  
Brazil

Received: August 26, 2022.  
Accepted: April 22, 2023.