Quantum Circuit Design using Complex valued Neural Network in Stiefel Manifold

Sayan Manna
Department Artificial Intelligence
IIT Kharagpur
sayan2003@kgpian.iitkgp.ac.in

Mahesh Mohan M R
Department of Artificial Intelligence
IIT Kharagpur
mahesh.mohan@cai.iitkgp.ac.in

Abstract

Quantum algorithms operate on quantum states through unitary transformations in high-dimensional complex Hilbert space. In this work, we propose a machine learning approach to create the quantum circuit using a single-layer complex-valued neural network. The input quantum state is provided to the network, which is trained to approximate the output state of a given quantum algorithm. To ensure that the fundamental property of unitarity is preserved throughout the training process, we employ optimization in Stiefel Manifold.

1 Introduction

Quantum computing exploits quantum mechanics to process information beyond the capabilities of classical computation [9]. A central research direction is the design of quantum algorithms with provable advantages over classical ones [5, 13]. These are typically expressed in the *quantum circuit model*, which are built from sequences of quantum gates acting on quantum states which are complex in nature [3, 12], enabling practical implementation on quantum devices or simulators.

Despite progress, mapping quantum algorithms to suitable gate sequences remains challenging, with many approaches [6, 1] still lacking efficient solutions. This motivates automated methods for generating task-specific quantum circuits. At the same time, machine learning—well-established for prediction and classification—has increasingly been combined with quantum computing in recent research [7].

In this paper, we propose a neural network-based approach for constructing quantum circuits. Existing methods such as Lie algebra approaches and Gram–Schmidt orthogonalization face limitations, as they cannot simultaneously guarantee unitarity preservation while ensuring a consistent decrease in the loss function. Here we have tried to solve this problem. Quantum evolution applies a unitary operator U to a quantum state, where U satisfies $UU^\dagger = I$, where $U^\dagger$ is conjugate transpose of U. Since quantum evolution is governed by unitary operators [10], we ensure that the network's weight matrices remain unitary during training, enabling a natural mapping to quantum circuits. Our method leverages optimization of weight matrix of complex valued neural network in *Stiefel manifold*, addressing limitations of existing approaches.

2 Problem Definition

Consider a quantum algorithm for which many input and output quantum states are known. Our objective is to design a neural network that can learn the corresponding unitary transformation of the quantum circuit. We construct a single-layer, complex-valued neural network for this purpose. For a circuit on n qubits, the state vector lies in $\mathbb{C}^{2^n}$ and evolves as $|\Psi'\rangle = U|\Psi\rangle$ with $U \in \mathbb{C}^{2^n \times 2^n}$. Thus, the neural network takes 2^n complex inputs and is trained to learn the unitary transformation between input and output states. Let $W \in \mathbb{C}^{N\times N}$, $N = 2^n$ be the network weight matrix, and $f(W)$ a real-valued loss. Training becomes the constrained optimization

$$\min_{W\in\mathbb{C}^{N\times N}} f(W) \quad \text{subject to} \quad W^\dagger W = I \qquad (1)$$

i.e., optimizing $f(W)$ while preserving the unitarity of W.

3 Literature Study

Enforcing unitarity in neural network weights has been studied in varius aspects [2, 8]. Use of machine learning to design scalable quantum circuit also has been studied in [11, 15]. Common approaches include Gram–Schmidt orthogonalization and Lie algebra parametrizations [17, 4], where weights are re-orthogonalized after fixed epochs. However, these methods only enforce unitarity intermittently, often causing abrupt increases in loss and yielding matrices that are not unitary, limiting their physical realizability. To address this, we adopt the *Cayley transform* on the Stiefel manifold, first proposed for RNN training [16], which preserves unitarity at every update. This ensures smooth loss decrease and strictly unitary weights throughout training. We extend this idea to quantum circuit design, representing one of the first applications of Cayley-transform-based manifold optimization in quantum computing.

4 Methodology

The core of this approach relies on optimizing neural network weights over the Stiefel manifold. The Stiefel manifold, denoted as $\mathcal{V}_N(\mathbb{C}^N)$, is defined as the set of all complex $N \times N$ matrices whose columns form an orthonormal basis in $\mathbb{C}^N$ [14], i.e.,

$$\mathcal{V}_N(\mathbb{C}^N) = \{W \in \mathbb{C}^{N\times N} \; : \; W^\dagger W = I_N\}. \quad (2)$$

Here, $W^\dagger$ represents the conjugate transpose of W, and I_N is the $N \times N$ identity matrix. For any $W \in \mathcal{V}_N(\mathbb{C}^N)$, we consider a loss function $f(W)$. The Euclidean gradient of f with respect to the entries of W is denoted by $G \in \mathbb{C}^{N\times N}$, where $G_{ij} = \frac{\partial f}{\partial W_{ij}}$. From this gradient, we construct a skew-Hermitian matrix A ($A^\dagger = -A$) where,

$$A = GW^\dagger - WG^\dagger \quad (3)$$

It defines a valid search direction tangent in the Stiefel manifold [16].

A descent curve along the manifold at training iteration k is then obtained by applying the Cayley transformation of $A^{(k)}$ to the current solution $W^{(k)}$. The update rule is expressed as:

$$W^{(k+1)} = \left(I + \frac{\lambda}{2}A^{(k)}\right)^{-1}\left(I - \frac{\lambda}{2}A^{(k)}\right)W^{(k)} \quad (4)$$

where $\lambda \in \mathbb{R}$ is the learning rate, chosen such that $\lambda > 0$ and sufficiently small. This formulation guarantees two important desired properties:

1. The updated weight matrix remains on the Stiefel manifold, i.e., $W^{(k+1)} \in \mathcal{V}_N(\mathbb{C}^N)$.

2. The loss function decreases globally, i.e., $f(W^{(k+1)}) \leq f(W^{(k)})$.

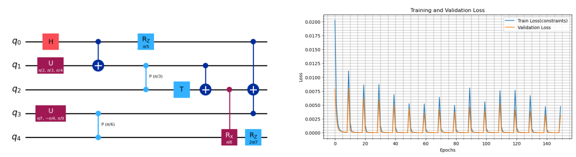
Please see the detailed proof of these properties in the Appendix.

We design a single-layer complex-valued neural network with weights initialized as a random unitary matrix ($W \in \mathcal{V}_N(\mathbb{C}^N)$). Training is performed using the prescribed update rule, yielding a final unitary matrix $W_{\text{final}} \in \mathcal{V}_N(\mathbb{C}^N)$. After the network training, we use the transpile function to convert the W_{final} into quantum circuits.

The `transpile` function converts a quantum circuit into an optimized form executable on specific hardware or simulators (e.g, Qiskit's `transpile` function).

5 Results and Discussion

We used a 5 qubit standard quantum circuit for experiment.

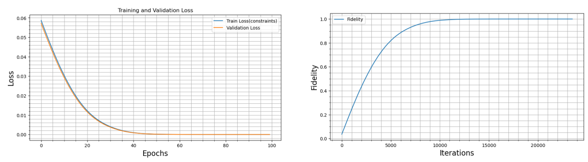


(a) 5-qubit sample quantum circuit

(b) Loss vs Epochs using Gram-Schmidt method

Figure 1: Sample circuit and Gram-Schmidt loss

As shown in Fig. 1b, the training loss increases whenever the Gram-Schmidt orthogonalization constraint is being applied to the weight matrix. Here unitarity is not being maintained during training.



(a) Loss vs Epochs using optimization on Steifel manifold

(b) Fidelity vs iteration on Stiefel manifold

Figure 2: Training loss and Fidelity

According to our approach as shown in Fig. 2a the loss is decreasing when Eq.(4) is being used. Throughout the training, the unitary error is observed to be of the order $\left\|WW^\dagger - I\right\|_2^2 \sim 10^{-11}$. This shows that unitarity is being maintained. We quantify the closeness between the learned unitary U_{learned} and the target unitary U_{true} using the fidelity

$$F = \frac{1}{d}\left|\text{Tr}\left(U_{\text{true}}^\dagger U_{\text{learned}}\right)\right| \quad (5)$$

where d is the dimension of the uniatry matrix. A value $F = 1$ indicates $U_{\text{learned}} = U_{\text{true}}$. As shown in Fig. 2b, the fidelity increases during training and finally reaches $F = 1$.

References

[1] Anonymous. 2021. The bitter truth about gate-based quantum algorithms in the nisq era. *Quantum Information Processing (IOP)*. Discusses how limited qubit connectivity forces additional gates and increases circuit depth.

[2] Martin Arjovsky, Amar Shah, and Yoshua Bengio. 2016. Unitary evolution recurrent neural networks. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1120–1128.

[3] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. 1995. Elementary gates for quantum computation. *Physical Review A*, 52(5):3457.

[4] Olivier Coulaud, Luc Giraud, and Martina Iannacito. 2022. On some orthogonalization schemes in tensor train format. arXiv preprint arXiv:2211.08770. Available at `https://arxiv.org/abs/2211.08770`.

[5] David Deutsch and Richard Jozsa. 1992. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907):553–558.

[6] Yan Ge, Wu Wenjie, Chen Yuheng, and 1 others. 2024. Quantum circuit synthesis and compilation optimization: Overview and prospects. *Preprint (arXiv)*. Provides a survey on translating algorithms into optimized gate sequences under NISQ constraints.

[7] Essam H. Houssein, Zainab Abohashima, Mohamed Elhoseny, and Waleed M. Mohamed. 2022. Machine learning in the quantum realm: The state-of-the-art, challenges, and future vision. *Expert Systems with Applications*, 194:116512.

[8] Zakaria Mhammedi, Andrew Hellicar, Ashfaqur Rahman, and James Bailey. 2016. Efficient orthogonal parametrisation of recurrent neural networks using householder reflections. *arXiv preprint arXiv:1612.00188*.

[9] Michael A. Nielsen and Isaac L. Chuang. 2002. *Quantum Computation and Quantum Information*. Cambridge University Press.

[10] Edward Parker. 2025. Unitary time evolution in quantum mechanics is a stronger physical postulate than linear time evolution. *Foundations of Physics*, 55:1–13.

[11] Rohit Sarma Sarkar and Bibhas Adhikari. 2024. A quantum neural network framework for scalable quantum circuit approximation of unitary matrices. Accessible at arXiv:2405.00012.

[12] Vivek V. Shende, Stephen S. Bullock, and Igor L. Markov. 2006. Synthesis of quantum-logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):1000–1010.

[13] Peter W. Shor. 1994. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134. IEEE.

[14] Hemant D. Tagare. 2011. Notes on optimization on stiefel manifolds. Technical report, Yale University.

[15] Kwok Ho Wan, Oscar Dahlsten, Hlér Kristjánsson, Robert Gardner, and M. S. Kim. 2016. Quantum generalisation of feedforward neural networks. *arXiv preprint arXiv:1612.01045*.

[16] Scott Wisdom, Thomas Powers, John R. Hershey, Jonathan Le Roux, and Les Atlas. 2016. Full-capacity unitary recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 4880–4888.

[17] M. Zomorodi, H. Amini, M. Abbaszadeh, J. Sohrabi, V. Salari, and P. Plawiak. 2024. Optimal quantum circuit design via unitary neural networks. arXiv preprint arXiv:2408.13211.

Appendix

Optimization in Stiefel Manifold

We consider the Stiefel manifold of all $N \times N$ complex-valued matrices whose columns are N orthonormal vectors in $\mathbb{C}^N$. It is defined as

$$\mathcal{V}_N(\mathbb{C}^N) = \{W \in \mathbb{C}^{N\times N} \mid W^\dagger W = I\}.$$

Tangent Space

A tangent space at a point on a manifold is the set of all possible directions in which we can move without leaving the manifold. Consider $W \in \mathcal{V}_N(\mathbb{C}^N)$. Since every point on the Stiefel manifold represents a unitary matrix, the tangent space at W, written as $\mathcal{T}_W \mathcal{V}_N(\mathbb{C}^N)$, contains all the directions (matrices) in which we can move infinitesimally while staying on the manifold.

The tangent space at W is defined as

$$\mathcal{T}_W \nu_N(\mathbb{C}^N) = \{Z \in \mathbb{C}^{N\times N} \mid W^\dagger Z + Z^\dagger W = 0\}. \tag{A1}$$

This condition ensures that if we move a small step ϵ to $W + \epsilon Z$, we still satisfy unitarity up to first order.

Proof

$$\begin{aligned}&(W+\epsilon Z)(W+\epsilon Z)^\dagger\\&=(W+\epsilon Z)(W^\dagger+\epsilon Z^\dagger)\\&=WW^\dagger+\epsilon ZW^\dagger+\epsilon WZ^\dagger+O(\epsilon^2)\\&=I+\epsilon\big(ZW^\dagger+WZ^\dagger\big)+O(\epsilon^2).\end{aligned}$$

Thus, if $ZW^\dagger + WZ^\dagger = 0$, we have $(W+\epsilon Z)(W+\epsilon Z)^\dagger = I$ up to first order, showing $Z \in \mathcal{T}_W\mathcal{V}_N(\mathbb{C}^N)$.

Skew-Hermitian Representation

Now, let A be any skew-Hermitian matrix, i.e., $A^\dagger = -A$. Then

$$Z = AW \in \mathcal{T}_W\mathcal{V}_N(\mathbb{C}^N) \tag{A2}$$

is a valid tangent vector. Indeed,

$$\begin{aligned}&(W+\epsilon Z)(W+\epsilon Z)^\dagger\\&=(W+\epsilon AW)(W+\epsilon AW)^\dagger\\&=(W+\epsilon AW)(W^\dagger-\epsilon W^\dagger A)\\&=WW^\dagger+\epsilon AWW^\dagger-\epsilon WW^\dagger A+O(\epsilon^2)\\&=I+\epsilon A-\epsilon A+O(\epsilon^2)\\&=I+O(\epsilon^2).\end{aligned}$$

Hence, $Z = AW$ indeed lies in the tangent space at W.

Canonical Inner Product and Riemannian Gradient

Next, we define the *canonical inner product*. The Stiefel manifold becomes a Riemannian manifold by introducing an inner product in its tangent spaces. Let $Z_1, Z_2 \in \mathcal{T}_W\mathcal{V}_N(\mathbb{C}^N)$, then the canonical inner product is given by

$$\langle Z_1, Z_2\rangle_c = \operatorname{tr}\left(Z_1^\dagger\left(I-\tfrac{1}{2}WW^{\mathrm{H}}\right)Z_2\right). \tag{A3}$$

Next, we define the gradients. Consider the smooth loss function

$$f(W) : \mathbb{C}^{N\times N} \to \mathbb{R}.$$

The Euclidean gradient $G \in \mathbb{C}^{N\times N}$ is defined as

$$G_{ij} = \frac{\partial f}{\partial W_{ij}}. \tag{A4}$$

If $Z \in \mathbb{C}^{N\times N}$, the differential of f, denoted by $Df(W)[Z] : \mathbb{C}^{N\times N} \to \mathbb{C}$, gives the derivative of f in the direction Z at W:

$$Df(W)[Z] = \operatorname{tr}(G^\dagger Z) = \sum_{i,j}\frac{\partial f}{\partial W_{ij}} Z_{ij}. \tag{A5}$$

Now we define the Riemannian gradient of f, denoted $\nabla_c f \in \mathcal{T}_W\mathcal{V}_N(\mathbb{C}^N)$. It is obtained as the projection of G onto the tangent space:

$$\nabla_c f = AW, \quad \text{where } A = GW^\dagger - WG^\dagger. \tag{A6}$$

Under the canonical inner product, the vector AW with $A = GW^\dagger - WG^\dagger$ represents the action of $Df(W)$ on the tangent space $\mathcal{T}_W\mathcal{V}_N(\mathbb{C}^N)$.

Consider the curve

$$Y(\lambda) = \left(I+\tfrac{\lambda}{2}A\right)^{-1}\left(I-\tfrac{\lambda}{2}A\right)W, \tag{A7}$$

where $W \in \mathcal{V}_N(\mathbb{C}^N)$, $G = \nabla f(W)$ is the Euclidean gradient of f, and $A = GW^\dagger - WG^\dagger$, $\lambda \in \mathbb{R}, \lambda > 0$ (sufficiently small).

Claim 1: $Y(\lambda)Y(\lambda)^\dagger = I$, i.e., $Y(\lambda) \in \mathcal{V}_N(\mathbb{C}^N)$.

Proof: Since

$$\begin{aligned}A^\dagger &= (GW^\dagger - WG^\dagger)^\dagger\\&= WG^\dagger - GW^\dagger\\&= -A,\end{aligned}$$

A is a skew-Hermitian matrix.

Now,

$$\begin{aligned}&Y(\lambda)^\dagger Y(\lambda)\\&=\Big(\big(I+\tfrac{\lambda}{2}A\big)^{-1}\big(I-\tfrac{\lambda}{2}A\big)W\Big)^\dagger\big(I+\tfrac{\lambda}{2}A\big)^{-1}\big(I-\tfrac{\lambda}{2}A\big)W\\&=W^\dagger\big(I-\tfrac{\lambda}{2}A\big)^\dagger\big((I+\tfrac{\lambda}{2}A)^{-1}\big)^\dagger\big(I+\tfrac{\lambda}{2}A\big)^{-1}\big(I-\tfrac{\lambda}{2}A\big)W\\&=W^\dagger\big(I+\tfrac{\lambda}{2}A\big)\big(I-\tfrac{\lambda}{2}A\big)^{-1}\big(I+\tfrac{\lambda}{2}A\big)^{-1}\big(I-\tfrac{\lambda}{2}A\big)W\\&=W^\dagger\big(I+\tfrac{\lambda}{2}A\big)\big((I+\tfrac{\lambda}{2}A)(I-\tfrac{\lambda}{2}A)\big)^{-1}\big(I-\tfrac{\lambda}{2}A\big)W\\&=W^\dagger\big(I+\tfrac{\lambda}{2}A\big)\big(I-\tfrac{\lambda^2}{4}A^2\big)^{-1}\big(I-\tfrac{\lambda}{2}A\big)W\\&=W^\dagger\big(I+\tfrac{\lambda}{2}A\big)(I+\tfrac{\lambda}{2}A)^{-1}(I-\tfrac{\lambda}{2}A)^{-1}\big(I-\tfrac{\lambda}{2}A\big)W\\&=W^\dagger IW\\&=W^\dagger W\\&=I.\end{aligned}$$

Therefore,

$$Y(\lambda)^\dagger Y(\lambda) = I \quad\Longrightarrow\quad Y(\lambda) \in \mathcal{V}_N(\mathbb{C}^N). \tag{A8}$$

Claim 2: For small enough λ, we have

$$f(Y(\lambda)) \le f(W).$$

Proof: We have

$$Y(\lambda) = \left(I + \tfrac{\lambda}{2}A\right)^{-1}\left(I - \tfrac{\lambda}{2}A\right)W.$$

At $\lambda = 0$, clearly $Y(0) = W$.

We know that if $M(\lambda)$ is invertible and differentiable, then

$$\frac{d}{d\lambda}M(\lambda)^{-1} = -M(\lambda)^{-1}\frac{dM(\lambda)}{d\lambda}M(\lambda)^{-1}. \tag{A9}$$

Now,

$$\begin{aligned} &Y'(\lambda) \\ &= \Bigg[\frac{d}{d\lambda}\left(I + \tfrac{\lambda}{2}A\right)^{-1}\left(I - \tfrac{\lambda}{2}A\right) + \\ &\qquad \left(I + \tfrac{\lambda}{2}A\right)^{-1}\frac{d}{d\lambda}\left(I - \tfrac{\lambda}{2}A\right)\Bigg]W \end{aligned} \tag{A10}$$

Using equation (A9), we obtain

$$Y'(0) = \left(-\tfrac{1}{2}A\right)IW + I\left(-\tfrac{1}{2}A\right)W = -AW$$

Thus, the tangent vector of the curve $Y(\lambda)$ at the starting point is

$$Y'(0) = -AW.$$

Now, under the canonical inner product on the tangent space, the gradient in the Stiefel manifold of the loss function f with respect to the matrix W is AW, where $A = GW^\dagger - WG^\dagger$. That is, for every tangent vector Z (or in the direction of Z),

$$Df(W)[Z] = \langle AW, Z\rangle_c \tag{A11}$$

Therefore, the directional derivative of f along the curve at $\lambda = 0$ is

$$\begin{aligned} \frac{d}{d\lambda}\Big|_{\lambda=0} f(Y(\lambda)) &= Df(W)[Y'(0)] \\ &= Df(W)[-AW] \\ &= \langle AW, -AW\rangle_c \\ &= -\langle AW, AW\rangle_c \\ &= -\|AW\|_c^2 \le 0 \end{aligned}$$

$$\frac{d}{d\lambda}\Big|_{\lambda=0} f(Y(\lambda)) = -\|AW\|_c^2 \le 0 \tag{A12}$$

Hence, the first directional derivative is strictly negative if the Riemannian gradient is non-zero (i.e., $AW \neq 0$).

Because f is smooth, for small $\lambda > 0$ we have the Taylor expansion with remainder,

$$f(Y(\lambda)) = f(W) + \lambda\frac{d}{d\lambda}\Big|_{\lambda=0} f(Y(\lambda)) + R(\lambda) \tag{A13}$$

where $R(\lambda) = O(\lambda^2)$.

More concretely, there exist $c > 0$ such that

$$|R(\lambda)| \le c\lambda^2 \tag{A14}$$

Therefore, for $\lambda > 0$, using (A12), we obtain

$$\begin{aligned} f(Y(\lambda)) &\le f(W) - \lambda\|AW\|_c^2 + c\lambda^2 \\ &= f(W) - \lambda\|AW\|_c^2\left(1 - \frac{c\lambda}{\|AW\|_c^2}\right) \end{aligned}$$

If $AW \neq 0$, pick $\lambda > 0$ small enough so that

$$\frac{c\lambda}{\|AW\|_c^2} \le \frac{1}{2}$$

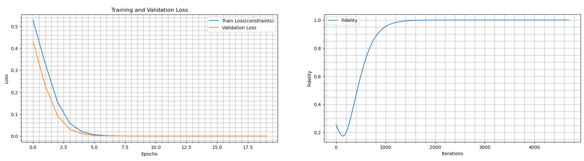
Then, for such λ,

$$f(Y(\lambda)) \le f(W) - \frac{\lambda}{2}\|AW\|_c^2 \le f(W) \tag{A15}$$

Thus, for all sufficiently small positive λ, the update rule decreases f unless $AW = 0$.

Additional Result: 2-qubit quantum circuit synthesis

In this example, we take 2-qubit entanglement operation. Now we want the quantum circuit for it using neural network. The input and output quantum states of the quantum operation is known. First, we create a complex valued neural network with 4 inputs and 4 outputs which ensures the dimension of unitary matrix to be 4×4. We train the network with Caley update rule and get the following results.



(a) Loss vs Epochs in steifel manifold

(b) Fidelity vs iteration in Stiefel Manifold

Figure 3: Training loss and Fidelity for a 2-qubit quantum operation

As we can see, the loss is decreasing and fidelity is reaching 1. Next, we extract the unitary weight matrix from the neural network and the obtained unitary error, $\left\|WW^{\dagger} - I\right\|_2^2 = 5.637545 \times 10^{-14}$ which is negligible. Then we transpile the unitary matrix with Qiskit's Transpile function to obtain the corresponding quantum circuit.

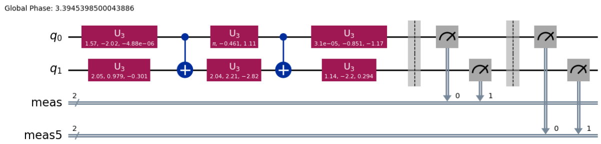


Figure 4: 2-qubit optimized quantum circuit after transpilation