J. Indones. Math. Soc.
Vol. 31, No. 02 (2025), pp. 1–16.

Representation Theory of Recurrent Neural Network

Lucky Cahya Wanditra[1], Intan Muchtadi Alamsyah[2,3*], Dellavitha Nasution[3]

[1]Doctoral Program in Mathematics, Institut Teknologi Bandung, Indonesia, 30122006@mahasiswa.itb.ac.id
[2]University Center of Excellence on Artificial Intelligence for Vision, Natural Language Processing and Big Data Analytics (U-CoE AI-VLB), Institut Teknologi Bandung, Indonesia, ntan@itb.ac.id
[3] Algebra Research Group, Institut Teknologi Bandung, Indonesia, dellavitha@itb.ac.id

Abstract. In this paper, we use the representation morphism concept to analyze the connection between two recurrent neural networks, primarily when we evaluate the neural network function between two isomorphic neural networks. We construct the set of all isomorphic classes of recurrent neural networks. We build the set by the action of the isomorphism group on the set of all recurrent neural networks that have invertible weight. By the group's action, we get the set of orbits and call it the moduli space. We analyze the moduli space to get its dimensions.

Key words and Phrases: quiver representation, recurrent neural network, recurrent neural network function, moduli spaces, dimensions

1. INTRODUCTION

Artificial intelligence is a technological breakthrough that impacts our lives. Artificial intelligence can help us solve many problems [1][2]. Because of this, some people try to understand how artificial intelligence works. The standard artificial intelligence algorithm is the artificial neural network. Artificial neural networks work by imitating the human brain. The algorithm uses some linear algebra to extract raw data to turn it into information. The algorithm uses the information to give the machine knowledge by machine learning. Currently, there are many machine-learning algorithms for artificial intelligence, among which artificial neural networks are the most common. Because artificial neural networks imitate how the human brain works, we can modify the algorithm to try to mimic how humans learn. Recurrent neural networks are one type of algorithm that try to imitate how humans learn [3].

Like humans, who learn from experience, machines can be made to learn from data history and experience. A recurrent neural network is an artificial algorithm using experiental data to minimize learning errors. This study tries to find some mathematical

2020 Mathematics Subject Classification: 16G20, 20C05, 20C35
Received: 14-10-2024, accepted: 21-06-2025.

properties of recurrent neural networks. Recurrent network quivers are used as a mathematical model of recurrent neural networks. Recurrent neural networks are defined using $\mathbb{C}^G$-representations of a recurrent network quiver. We will find some relations between two neural networks using the representation theory concepts. Furthermore, the isomorphism relation between two neural networks is discovered. A moduli space of recurrent neural networks is created from the isomorphism relation and the dimensions are found. From the dimensions of the moduli space, the complexity of the neural network can be seen, primarily if the neural network uses $ReLu(x) = max(0, x)$ as an activation function. Furthermore, neural teleportation for recurrent neural networks with higher dimensions is defined. Neural teleportation is a neural network algorithm that can jump processes from one layer to another [4]. This jump will increase the algorithm's speed because it needs fewer layers to get the predictions.

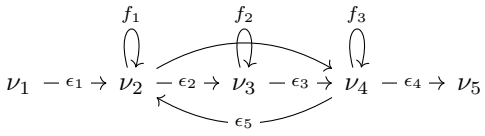

FIGURE 1. Example of Neural Teleportation

2. Quiver Representation and Group Algebra

2.1. **Quiver Representation.**

Let $\mathcal{Q} = (\mathcal{V}, \mathcal{E}, s, t)$ be a quiver where $\mathcal{V}$ is a set of vertices, $\mathcal{E}$ is a set of arrows, and $s, t : \mathcal{E} \to \mathcal{V}$ map every arrow $\epsilon \in \mathcal{E}$ to its source vertex $s(\epsilon) \in V$ and target vertex $t(\epsilon) \in \mathcal{V}$, respectively. A quiver can have loops[5]. An arrow $\epsilon \in \mathcal{E}$ is a loop if $s(\epsilon) = t(\epsilon)$.

Definition 2.1. [6] *A quiver $\mathcal{Q} = (\mathcal{V}, \mathcal{E}, s, t)$ is arranged by layers if it can be drawn from left to right, arranging its vertices in columns such that:*

(1) there are no oriented edges from vertices on the right to vertices on the left,
(2) for every $\epsilon \in \mathcal{E}$, we have $s(\epsilon) = t(\epsilon)$ if and only if $s(\epsilon), t(\epsilon)$ in the same column .

We can enumerate every column in a quiver that is arranged by layer. The first layer is the left-most column and is called the input layer. The last layer is the right-most column and is called the output layer. Columns between the input and output layers are called hidden layers. The hidden layers are enumerated from left to right (the first hidden layer, the second hidden layer, etc.). Vertices in the input layer are called input vertices, while vertices in the output layer are called output vertices. A vertex v is called a bias vertex if ν is in a hidden layer and for all $\epsilon \in \mathcal{E}$, $t(\epsilon) \neq \nu$. A vertex v is called a hidden vertex if ν is in a hidden layer and is not a bias vertex.

Definition 2.2. [6] *A quiver $\mathcal{Q}$ is called a network quiver if it satisfies the following conditions:*

(1) $\mathcal{Q}$ is arranged by layers,
(2) every input, output, and bias vertex does not have any loop,

(3) every hidden vertex has exactly one loop .

A delooped quiver $\mathcal{Q}^\circ = (\mathcal{V}, \mathcal{E}^\circ, s^\circ, t^\circ)$ of Q is a quiver Q from which all loops of $\mathcal{Q}$ have been removed[6].

Let $\mathcal{Q} = (\mathcal{V}, \mathcal{E}, s, t)$ be a quiver. A representation of a quiver $\mathcal{Q}$ is defined by a tuple $\mathrm{W} = (\{\mathrm{W}_\nu\}_{\nu\in\mathcal{V}}, \{\mathrm{W}_\epsilon\}_{\epsilon\in\mathcal{E}})$, where $\{\mathrm{W}_\nu\}_{\nu\in\mathcal{V}}$ is a sequence of vector spaces that is indexed by $\nu \in \mathcal{V}$ and $\{\mathrm{W}_\epsilon\}_{\epsilon\in\mathcal{E}}$ is a sequence of linear transformations that is indexed by $\epsilon \in \mathcal{E}$ such that for every $\epsilon \in \mathcal{E}$ we have

$$\mathrm{W}_\epsilon : \mathrm{W}_{s(\epsilon)} \to \mathrm{W}_{t(\epsilon)}.$$

Let U, W be two representations of $\mathcal{Q}$. A morphism representation $\tau : \mathrm{W} \to \mathrm{U}$ is a sequence of linear transformations $\tau = \{\tau_\nu : \mathrm{U}_\nu \to \mathrm{W}_\nu\}_{\nu\in\mathcal{V}}$ that is indexed by $\nu \in \mathcal{V}$ such that for every $\epsilon \in \mathcal{E}$ we have

$$\tau_{t(\epsilon)}\mathrm{W}_\varepsilon = \mathrm{U}_\epsilon\tau_{s(\epsilon)}.$$

A morphism τ is called an isomorphism if τ_ν is invertible for every $\nu \in \mathcal{V}$. We say W is isomorphic to U if τ is an isomorphism [6].

2.2. **Convolution Representation.**

Let G be a finite group and define

$$\mathbb{C}^G = \{f : G \to \mathbb{C}\}$$

as a set of functions from the group G to the complex number set $\mathbb{C}$, We define the addition, scalar multiplication, and inner product in $\mathbb{C}^G$, respectively, as follows:

(1) for every $f, g \in \mathbb{C}^G$ and $x \in G$, we have $(f + g)(x) = f(x) + g(x)$,
(2) for every $f \in \mathbb{C}^G$, $z \in \mathbb{C}$, and $x \in G$, we have $(zf)(x) = zf(x)$,
(3) for every $f, g \in \mathbb{C}^G$, we can define $\langle f, g\rangle = \sum_{x\in G} f(x)\overline{g(x)}$; $\overline{g(x)}$ as the complex conjugate of $g(x)$.

We know that $\mathbb{C}^G$ forms a vector space. We will define the pointwise multiplication $(\cdot)$ operation in $\mathbb{C}^G$. Let $f, g \in \mathbb{C}^G$ and define

$$(f \cdot g)(x) = f(x)g(x); \forall x \in \mathbb{C}.$$

We can say that $\mathbb{C}^G$ is an algebra over $\mathbb{C}$, because $\mathbb{C}^G$ is a vector space over $\mathbb{C}$ and a ring with pointwise multiplication; we call $\mathbb{C}^G$ a group algebra. We also define a convolution operation in $\mathbb{C}^G$ as follows:

$$(f * g)(x) = \sum_{y\in G} f(y)g(y^{-1}x)$$

for every $f, g \in \mathbb{C}^G$ and $x \in G$ [7].

The mathematical system $(\mathbb{C}^G, +, *)$ also forms a ring. Hence, $\mathbb{C}^G$ can be considered as an algebra under the convolution operation [7].

We will consider $\mathbb{C}^G$ as a vector space and use it to get a quiver representation. Let $\mathcal{Q} = (\mathcal{V}, \mathcal{E}, s, t)$ be a quiver. We can define a quiver representation $(\{\mathbb{C}^G\}_{\nu\in\mathcal{V}}, \{W_\epsilon\}_{\epsilon\in\mathcal{E}})$, where W_ϵ is a linear map from $\mathbb{C}^G$ to $\mathbb{C}^G$. The representation is called the $\mathbb{C}^G$-representation of $\mathcal{Q}$ [8].

Furthermore, we also can define the $\mathbb{C}^G$-convolution representation of $\mathcal{Q}$. The $\mathbb{C}^G$-convolution representation of $\mathcal{Q}$ is a $\mathbb{C}^G$-representation of $\mathcal{Q}$ such that for every $\epsilon \in \mathcal{E}$ there is a $w_\epsilon \in \mathbb{C}^G$ such that $W_\epsilon(a) = w_\epsilon * a$ for every $a \in \mathbb{C}^G$ [8].

3. Recurrent Neural Network

Definition 3.1. [6] *The hidden quiver of $\mathcal{Q}$, denoted by $\tilde{\mathcal{Q}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}}, \tilde{s}, \tilde{t})$, is defined by the hidden vertices $\tilde{\mathcal{V}}$ of $\mathcal{Q}$ and all arrows $\tilde{\mathcal{E}}$ between the hidden vertices of $\mathcal{Q}$ that are not loops. We will refer to the layers in $\tilde{\mathcal{Q}}$ as hidden layers.*

Definition 3.2. [8] *A recurrent network quiver $\mathcal{Q}$ is a network quiver $\mathcal{Q}$ augmented by arrows from all vertices in the last hidden layer to all vertices in the first hidden layer.*

Definition 3.3. [8] *A $\mathbb{C}^G$-recurrent neural network over a recurrent network quiver Q is a pair (W, f) where W is a $\mathbb{C}^G$-representation of delooped quiver Q° and $f = \{f_v\}_{v \in \tilde{V}}$ is a sequence of differentiable functions from $\mathbb{C}^G$ to $\mathbb{C}^G$. The function $f_v : \mathbb{C}^G \to \mathbb{C}^G$ is called the activation function of vertex ν.*

Let $\zeta_\nu = \{\epsilon \in E | t(\epsilon) = \nu\}$. We can also define $\hat{\zeta}_\nu = \{\epsilon \in \zeta_\nu | s(\epsilon) \text{ in the right layer}\}$ and $\bar{\zeta}_\nu = \{\epsilon \in \zeta_\nu | s(\epsilon) \text{ in the left layer}\}$. Therefore, we can get that $\zeta_\nu = \hat{\zeta}_\nu \bigcup \bar{\zeta}_\nu$ and $\hat{\zeta}_\nu \bigcap \bar{\zeta}_\nu = \emptyset$. Furthermore, we know that $\hat{\zeta}_\nu$ is $\emptyset$ if ν is not in the first hidden layer, which implies that

$$\zeta_\nu = \begin{cases} \hat{\zeta}_\nu \bigcup \bar{\zeta}_\nu & ; \text{if } \nu \text{ is in the first hidden layer,} \\ \bar{\zeta}_\nu & ; \text{if } \nu \text{ is not in the first hidden layer.} \end{cases}$$

Let $\mathcal{Q}$ be a $\mathbb{C}^G$-recurrent network quiver. Let d be the number of input vertices of $\mathcal{Q}$. Let (W, f) be a $\mathbb{C}^G$-recurrent neural network over a recurrent network quiver $\mathcal{Q}$. We will define a sequence of functions $a_n(\mathrm{W}, f)_v : (\mathbb{C}^G)^d \to \mathbb{C}^G$ that maps an input data $x = \{x_\nu\}_{\nu \in \mathcal{V}_{in}} \in (\mathbb{C}^G)^d$ to the output value of a vertex in $\mathcal{Q}$ where:

$$a_n(\mathrm{W}, f)_\nu(x) = \begin{cases} x_\nu & ; \text{if } \nu \text{ is an input vertex,} \\ 1 & ; \text{if } \nu \text{ is a bias vertex,} \\ f\left(\sum_{\delta \in \hat{\zeta}_\nu} \mathrm{W}_\epsilon a_{n-1}(\mathrm{W}, f)_{s(\delta)}(x) \right. & \\ \left. + \sum_{\epsilon \in \bar{\zeta}_\nu} \mathrm{W}_\epsilon a_n(\mathrm{W}, f)_{s(\epsilon)}(x)\right) & ; \text{if } \nu \text{ is a hidden vertex,} \\ \sum_{\epsilon \in \zeta_\nu} \mathrm{W}_\epsilon a_n(\mathrm{W}, f)_{s(\epsilon)}(x) & ; \text{if } \nu \text{ is an output vertex,} \end{cases}$$

and $a_0(\mathrm{W}, f)_\nu(x) = 0$ for every $\nu \in \mathcal{V}$ [8].

Definition 3.4. [8] *Let (W, f) be a $\mathbb{C}^G$-recurrent neural network over a recurrent network quiver $\mathcal{Q}$. Let d be the number of input vertices of $\mathcal{Q}$ and k be the number of output vertices of $\mathcal{Q}$. Let us now define the recurrent neural network function as follows:*

$$\begin{aligned} \Psi(\mathrm{W}, f) : (\mathbb{C}^G)^d &\to l((\mathbb{C}^G)^k) \\ x &\mapsto \{a_n(\mathrm{W}, f)_\nu(x)\}_{\nu \in \mathcal{V}_{out}} \end{aligned}$$

where $\mathcal{V}_{out}$ is the set of all vertices in the output layer and $l((\mathbb{C}^G)^k)$ is the set of all sequences of $(\mathbb{C}^G)^k$.

Definition 3.5. [8] *Let* (W, f) *and* (U, g) *be two* $\mathbb{C}^G$*-recurrent neural networks over a network quiver* $\mathcal{Q}$*. A morphism of* $\mathbb{C}^G$*-recurrent neural networks is a morphism of* $\mathbb{C}^G$*-representations* $\tau : (\mathrm{W}, f) \to (\mathrm{U}, g)$ *that satisfies*

(1) $\tau_\nu = 1$ *for every* $\nu \notin \tilde{\mathcal{V}}$*;*
(2) for every $\nu \in \tilde{\mathcal{V}}$*, we have this commutative diagram:*

$$\begin{array}{ccc} \mathbb{C}^G & \xrightarrow{f_\nu} & \mathbb{C}^G \\ \downarrow{\scriptstyle \tau_\nu} & & \downarrow{\scriptstyle \tau_\nu} \\ \mathbb{C}^G & \xrightarrow{g_\nu} & \mathbb{C}^G \end{array}$$

FIGURE 2. Commutative Diagram for Morphism of $\mathbb{C}^G$-recurrent neural networks

where $\tilde{\mathcal{V}}$ *is the set of all vertices in the hidden layer. If for every* $\nu \in \mathcal{V}$*,* τ_ν *is bijective, then* τ *is an isomorphism and* (W, f) *is isomorphic to* (U, g) *.*

Lemma 3.6. [8] *Let* $\mathcal{Q} = (\mathcal{V}, \mathcal{E}, s, t)$ *be a recurrent network quiver,* G *be a finite group. Define*

$$\mathcal{C}_\Gamma(\mathcal{Q}) = \{I\}_{\nu \in V_{in}} \times \prod_{v \in \tilde{V}} GL(\mathbb{C}^G) \times \{I\}_{v \in V_{out}}$$

where $GL(\mathbb{C}^G)$ *is a general linear group of* $\mathbb{C}^G$*. Then* $\mathcal{C}_\Gamma(\mathcal{Q})$ *is the group of all isomorphism of* $\mathbb{C}^G$*-recurrent neural networks over recurrent network quiver* $\mathcal{Q}$*.*

Proof. It is clear that $\mathcal{C}_\Gamma(\mathcal{Q})$ is a group. Let $\tau \in \mathcal{C}_\Gamma(\mathcal{Q})$. Let (W, f) is a recurrent neural network over network quiver $\mathcal{Q}$. Define

$$(U, g) = (\tau W \tau^{-1}, \tau g \tau^{-1}) = (\{\tau_{t(\epsilon)} W_\epsilon \tau_{s(\epsilon}^{-1}\}_{\epsilon \in \mathcal{E}}, \{\tau_v g \tau_v^{-1}\}_{\nu \in \tilde{\mathcal{V}}})$$

where $\tilde{\mathcal{V}}$ is the set of all vertex in hidden quiver $\tilde{\mathcal{Q}}$ of recurrent network quiver $\mathcal{Q}$. From the definition of (U, g), we get that τ is a morphism of $\mathbb{C}^G$-recurrent neural network between (W, f) and (U, g). Therfore, $\mathcal{C}_\Gamma(\mathcal{Q})$ is the group of all isomorphism of $\mathbb{C}^G$-recurrent neural network over network quiver $\mathcal{Q}$. □

Theorem 3.7. *If* (W, f) *and* (U, g) *are two isomorphic* $\mathbb{C}^G$*-recurrent neural networks then*

$$\Psi(\mathrm{W}, f)(x) = \Psi(\mathrm{U}, g)(x).$$

Proof. Let (W, f) and (U, g) be two isomorphic $\mathbb{C}^G$-recurrent neural networks. This means there is an isomorphism τ such that $\tau \diamond (\mathrm{W}, f) = (\mathrm{V}, g)$. Let $x \in (\mathbb{C}^G)^d$ be an input vertex for (W, f) and (U, g). For $n = 0$, we know that $a_0(\mathrm{W}, f)_v(x) = 0 = a_0(\mathrm{U}, g)_v(x)$ for every vertex v in quiver $\mathcal{Q}$. For $n = 1$, we have

$$a_1(\mathrm{W}, f)_v(x) = a_1(\mathrm{U}, g)_v(x) = \begin{cases} x_\nu & ; \nu \text{ is an input vertex,} \\ 1 & ; \nu \text{ is a bias vertex,} \end{cases}$$

because $\tau_v = 1$. If v is in the first hidden layer, we have

$$
\begin{aligned}
a_1(\mathrm{U},g)_\nu(x) &= g_\nu\left(\sum_{\delta\in\hat{\zeta}_\nu}\mathrm{U}_\delta a_0(\mathrm{U},g)_{s(\delta)}(x)+\sum_{\epsilon\in\bar{\zeta}_v}\mathrm{U}_\epsilon a_1(\mathrm{U},g)_{s(\epsilon)}(x)\right)\\
&= g_v\left(0+\sum_{\epsilon\in\bar{\zeta}_v}\mathrm{U}_\epsilon a_1(\mathrm{U},g)_{s(\epsilon)}(x)\right)\\
&= \tau_\nu f_\nu\left(\tau_\nu^{-1}\sum_{\epsilon\in\bar{\zeta}_\nu}\mathrm{U}_\epsilon a_1(\mathrm{U},g)_{s(\epsilon)}(x)\right)\\
&= \tau_\nu f_\nu\left(\tau_\nu^{-1}\sum_{\epsilon\in\zeta_\nu}\tau_{t(\epsilon)}\mathrm{W}_\epsilon\tau_{s(\epsilon)}^{-1}a_1(\mathrm{U},g)_{s(\epsilon)^{-1}}(x)\right)\\
&= \tau_\nu f_\nu\left(\sum_{\epsilon\in\zeta_\nu}\tau_\nu^{-1}\tau_\nu\mathrm{W}_\epsilon a_1(\mathrm{U},g)_{s(\epsilon)}(x)\right) && ;\text{since } \tau_{s(\epsilon)}=1=\tau_{s(\epsilon)}^{-1}\\
&= \tau_v f_v\left(\sum_{\epsilon\in\zeta_v}\mathrm{W}_\epsilon a_1(\mathrm{U},g)_{s(\epsilon)}(x)\right).
\end{aligned}
$$

Since $s(\epsilon)$ is an input vertex, then $a_1(\mathrm{U},g)_{s(\epsilon)}(x)=a_1(\mathrm{W},f)_{s(\epsilon)}(x)$. Hence, we get

$$
a_1(\mathrm{U},g)_v(x)=\tau_\nu f_\nu\left(\sum_{\epsilon\in\zeta_\nu}\mathrm{W}_\epsilon a_1(\mathrm{W},f)_{s(\epsilon)}(x)\right).
$$

Therefore, we get

$$
a_1(\mathrm{U},g)_v(x)=\tau_v a_1(\mathrm{W},f)_v(x). \tag{1}
$$

For ν in other hidden layers, we will have $\zeta_\nu=\bar{\zeta}_\nu$ from the definition of a recurrent neural network. If ν is in the second hidden layer, we have

$$
\begin{aligned}
a_1(\mathrm{U},g)_\nu(x)(x) &= g_\nu\left(\sum_{\delta\in\hat{\zeta}_\nu}\mathrm{U}_\delta a_0(\mathrm{U},g)_{s(\delta)}(x)+\sum_{\epsilon\in\bar{\zeta}_\nu}\nu_\epsilon a_1(\mathrm{U},g)_{s(\epsilon)}(x)\right)\\
&= g_\nu\left(\sum_{\epsilon\in\zeta_\nu}\mathrm{U}_\epsilon a_1(\mathrm{U},g)_{s(\epsilon)}(x)\right) && ;\text{from the definition of recurrent neural network}\\
&= \tau_\nu f_\nu\left(\tau_\nu^{-1}\sum_{\epsilon\in\zeta_\nu}\mathrm{U}_\epsilon a_1(\mathrm{U},g)_{s(\epsilon)}(x)\right)\\
&= \tau_\nu f_\nu\left(\tau_\nu^{-1}\sum_{\epsilon\in\zeta_\nu}\tau_{t(\epsilon)}\mathrm{W}_\epsilon\tau_{s(\epsilon)}^{-1}a_1(\mathrm{U},g)_{s(\epsilon)}(x)\right)
\end{aligned}
$$

$$
\begin{aligned}
&= \tau_\nu f_\nu \left(\sum_{\epsilon \in \zeta_\nu} \tau_\nu^{-1} \tau_\nu \mathrm{W}_\epsilon \tau_{s(\epsilon)}^{-1} \tau_{s(\epsilon)} a_n(\mathrm{W}, f)_{s(\epsilon)}(x) \right) && ; \text{since } \epsilon \in \zeta_\nu \\
&= \tau_\nu f_\nu \left(\sum_{\epsilon \in \zeta_\nu} \mathrm{W}_\epsilon a_1(\mathrm{W}, f)_{s(\epsilon)}(x) \right) \\
&= \tau_\nu a_1(\mathrm{W}, f)_\nu(x) && ; \text{from the definition of } (\mathrm{W}, f).
\end{aligned}
$$

This means that if ν is in the second hidden layer, we get

$$a_1(\mathrm{U}, g)_\nu(x) = \tau_\nu a_1(\mathrm{W}, f)_\nu(x).$$

Inductively, we get $a_1(\mathrm{U}, g)_\nu(x) = \tau_\nu a_1(\mathrm{W}, f)_\nu(x)$ for every ν in a hidden layer. If ν is an output vertex, we have

$$
\begin{aligned}
a_1(\mathrm{U}, g)_\nu(x) &= \sum_{\epsilon \in \zeta_\nu} \mathrm{U}_\epsilon a_1(\mathrm{U}, g)_{s(\epsilon)}(x) \\
&= \sum_{\epsilon \in \zeta_\nu} \tau_{t(\epsilon)} \mathrm{W}_\epsilon \tau_{s(\epsilon)} a_1(\mathrm{U}, g)_{s(\epsilon)}(x) \\
&= \sum_{\epsilon \in \zeta_\nu} \tau_\nu \mathrm{W}_\epsilon \tau_{s(\epsilon)}^{-1} \tau_{s(\epsilon)} a_1(\mathrm{W}, f)_{s(\epsilon)}(x) && ; \text{since } s(\epsilon) \text{ is in a hidden layer} \\
&= \sum_{\epsilon \in \zeta_\nu} \mathrm{W}_\epsilon a_1(\mathrm{W}, f)_{s(\epsilon)}(x) && ; \text{since } \tau_\nu = 1 \text{ for } \nu \in \mathcal{V}_{out}.
\end{aligned}
$$

Therefore, we get

$$a_1(\mathrm{U}, g)_\nu(x) = a_1(\mathrm{W}, f)_\nu(x).$$

For $n = 2, 3, \cdots$, we have

$$a_n(\mathrm{W}, f)_\nu(x) = a_n(\mathrm{U}, g)_\nu(x) = \begin{cases} x_\nu & ; \nu \text{ is an input vertex} \\ 1 & ; \nu \text{ is a bias vertex} \end{cases}$$

because $\tau_\nu = 1$. If ν is in the first hidden layer, we have

$$
\begin{aligned}
a_n(\mathrm{U}, g)_\nu(x) &= g_\nu \left(\sum_{\delta \in \hat{\zeta}_\nu} \mathrm{U}_\delta a_{n-1}(\mathrm{U}, g)_{s(\delta)}(x) + \sum_{\epsilon \in \bar{\zeta}_\nu} \mathrm{U}_\epsilon a_n(\mathrm{U}, g)_{s(\epsilon)}(x) \right) \\
&= \tau_\nu f_\nu \left(\tau_\nu^{-1} \left(\sum_{\delta \in \hat{\zeta}_\nu} \mathrm{U}_\delta a_{n-1}(\mathrm{U}, g)_{s(\delta)}(x) + \sum_{\epsilon \in \bar{\zeta}_\nu} \mathrm{U}_\epsilon a_n(\mathrm{U}, g)_{s(\epsilon)}(x) \right) \right) \\
&= \tau_\nu f_\nu \left(\tau_\nu^{-1} \left(\sum_{\delta \in \hat{\zeta}_\nu} \tau_{t(\delta)} \mathrm{W}_\delta \tau_{s(\delta)}^{-1} a_{n-1}(\mathrm{U}, g)_{s(\delta)}(x) \right. \right.
\end{aligned}
$$

$$
\begin{aligned}
&+ \left. + \sum_{\epsilon \in \bar{\zeta}_v} \tau_{t(\epsilon)} \mathrm{W}_\epsilon \tau_{s(\epsilon)}^{-1} a_n(\mathrm{U}, g)_{s(\epsilon)}(x) \right) \Bigg) \\
&= \tau_\nu f_\nu \left(\tau_\nu^{-1} \left(\sum_{\delta \in \hat{\zeta}_\nu} \tau_\nu \mathrm{W}_\delta \tau_{s(\delta)}^{-1} \tau_{s(\delta)} a_{n-1}(\mathrm{W}, f)_{s(\delta)}(x) \right.\right. \\
&+ \left. \left. + \sum_{\epsilon \in \zeta_\nu} \tau_\nu \mathrm{W}_\epsilon \tau_{s(\epsilon)}^{-1} a_n(\mathrm{U}, g)_{s(\epsilon)}(x) \right) \right) \\
&= \tau_\nu f_\nu \left(\tau_\nu^{-1} \tau_\nu \left(\sum_{\delta \in \hat{\zeta}_\nu} \mathrm{W}_\delta \tau_{s(\delta)}^{-1} \tau_{s(\delta)} a_{n-1}(\mathrm{W}, f)_{s(\delta)}(x) \right.\right. \\
&+ \left. \left. \sum_{\epsilon \in \zeta_\nu} \mathrm{W}_\epsilon a_n(\mathrm{U}, g)_{s(\epsilon)}(x) \right) \right) \qquad \text{; since } \tau_{s(\epsilon)} = 1 \\
&= \tau_\nu f_\nu \left(\sum_{\delta \in \hat{\zeta}_\nu} \mathrm{W}_\delta a_{n-1}(\mathrm{W}, f)_{s(\delta)}(x) + \sum_{\epsilon \in \zeta_\nu} \mathrm{W}_\epsilon a_n(\mathrm{U}, g)_{s(\epsilon)}(x) \right).
\end{aligned}
$$

Since $s(\epsilon)$ is an input vertex, then $a_n(\mathrm{U}, g)_{s(\epsilon)}(x) = a_n(\mathrm{W}, f)_{s(\epsilon)}(x)$. Hence, we get

$$
a_n(\mathrm{U}, g)_\nu(x) = \tau_\nu f_\nu \left(\sum_{\delta \in \hat{\zeta}_\nu} \mathrm{W}_\delta a_{n-1}(\mathrm{W}, f)_{s(\delta)}(x) + \sum_{\epsilon \in \zeta_v} \mathrm{W}_\epsilon a_n(\mathrm{W}, f)_{s(\epsilon)}(x) \right).
$$

Therefore, we get

$$
a_n(\mathrm{U}, g)_\nu(x) = \tau_\nu a_n(\mathrm{W}, f)_\nu(x). \tag{2}
$$

For ν is in other hidden layers, we will have $\zeta_\nu = \bar{\zeta}_\nu$ from the definition of a $\mathbb{C}^G$-recurrent neural network. So, we have

$$
\begin{aligned}
a_n(\mathrm{U}, g)_\nu(x)(x) &= g_\nu \left(\sum_{\delta \in \hat{\zeta}_\nu} \mathrm{U}_\delta a_{n-1}(\mathrm{U}, g)_{s(\delta)}(x) + \sum_{\epsilon \in \bar{\zeta}_\nu} \mathrm{U}_\epsilon a_n(\mathrm{U}, g)_{s(\epsilon)}(x) \right) \\
&= g_\nu \left(\sum_{\epsilon \in \zeta_\nu} \mathrm{U}_\epsilon a_n(\mathrm{U}, g)_{s(\epsilon)}(x) \right) \qquad \text{; from the definition of} \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{recurrent neural network} \\
&= \tau_\nu f_\nu \left(\tau_\nu^{-1} \sum_{\epsilon \in \zeta_\nu} \mathrm{U}_\epsilon a_n(\mathrm{U}, g)_{s(\epsilon)}(x) \right) \\
&= \tau_\nu f_\nu \left(\tau_\nu^{-1} \sum_{\epsilon \in \zeta_\nu} \tau_{t(\epsilon)} \mathrm{W}_\epsilon \tau_{s(\epsilon)}^{-1} a_n(\mathrm{U}, g)_{s(\epsilon)}(x) \right)
\end{aligned}
$$

$$= \tau_\nu f_\nu \left(\sum_{\epsilon \in \zeta_\nu} \tau_\nu^{-1} \tau_\nu \mathrm{W}_\epsilon \tau_{s(\epsilon)}^{-1} \tau_{s(\epsilon)} a_n(\mathrm{W}, f)_{s(\epsilon)}(x) \right) \qquad ; \text{since } \epsilon \in \zeta_\nu$$

$$= \tau_\nu f_\nu \left(\sum_{\epsilon \in \zeta_\nu} \mathrm{W}_\epsilon a_n(\mathrm{W}, f)_{s(\epsilon)}(x) \right)$$

$$= \tau_\nu a_n(\mathrm{W}, f)_v(x) \qquad ; \text{from the definition of } (\mathrm{W}, f).$$

So, we get that

$$a_n(\mathrm{U}, g)_\nu(x) = \tau_\nu a_n(\mathrm{W}, f)_\nu(x)$$

for every ν in a hidden layer. For ν is an output layer, we will get

$$\begin{aligned} a_n(\mathrm{U}, g)_\nu(x) &= \sum_{\epsilon \in \zeta_\nu} \mathrm{U}_\epsilon a_n(\mathrm{U}, g)_{s(\epsilon)}(x) \\ &= \sum_{\epsilon \in \zeta_\nu} \tau_{t(\epsilon)} \mathrm{W}_\epsilon \tau_{s(\epsilon)} a_n(\mathrm{U}, g)_{s(\epsilon)}(x) \\ &= \sum_{\epsilon \in \zeta_\nu} \tau_\nu \mathrm{W}_\epsilon \tau_{s(\epsilon)}^{-1} \tau_{s(\epsilon)} a_n(W, f)_{s(\epsilon)}(x) \quad ; \text{since } s(\epsilon) \text{ is in a hidden layer} \\ &= \sum_{\epsilon \in \zeta_\nu} \mathrm{W}_\epsilon a_n(\mathrm{W}, f)_{s(\epsilon)}(x) \qquad ; \text{since } \tau_\nu = 1 \text{ for } \nu \in \mathcal{V}_{out} \end{aligned}$$

Therefore, we get

$$a_n(\mathrm{U}, g)_\nu(x) = a_n(\mathrm{W}, f)_\nu(x).$$

This means we get

$$\Psi(\mathrm{W}, f) = \Psi(\mathrm{U}, g).$$

□

Remark 3.1. *This theorem tells us that if we have two neural networks over the same quiver and they are isomorphic, then we can use a neural network that is simpler for calculation and more efficient in memory usage* [6],[4].

Example 3.8. Let $\mathcal{Q}$ be a recurrent network quiver that can be drawn as follows:

Let $G = \mathbb{Z}_2$ and $(\mathcal{W}, f)$ be a $\mathbb{C}^G$-representation of $\mathcal{Q}$ with

$$\mathrm{W}_{\epsilon_1} = \begin{pmatrix} 1 & 3 \\ 4 & 6 \end{pmatrix}, \quad \mathrm{W}_{\epsilon_2} = \begin{pmatrix} 1 & 5 \\ 7 & 4 \end{pmatrix}, \quad \mathrm{W}_{\epsilon_3} = \begin{pmatrix} 4 & 3 \\ 2 & 2 \end{pmatrix}, \quad \mathrm{W}_{\epsilon_4} = \begin{pmatrix} 1 & 1 \\ 3 & 2 \end{pmatrix},$$

$$\mathrm{W}_{\epsilon_5} = \begin{pmatrix} 4 & 1 \\ 0 & 1 \end{pmatrix}, \quad \mathrm{W}_{\epsilon_6} = \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}, \quad \mathrm{W}_{\epsilon_7} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \quad \mathrm{W}_{\epsilon_8} = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix},$$

$$\mathrm{W}_{\epsilon_9} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \mathrm{W}_{\epsilon_{10}} = \begin{pmatrix} 0 & 1 \\ 2 & 0 \end{pmatrix}, \quad \mathrm{W}_{\epsilon_{11}} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \quad \mathrm{W}_{\epsilon_{12}} = \begin{pmatrix} 3 & 0 \\ 1 & 2 \end{pmatrix}.$$

Let $\tau = \{\tau_v : \mathrm{W}_v \mapsto U_v\}_{v \in V}$ be a morphism representation of Q that induces a morphism of a recurrent neural network from (W, f) to (U, g) over recurrent network quiver Q with

$$\tau_{\nu_1} = \tau_{\nu_6} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

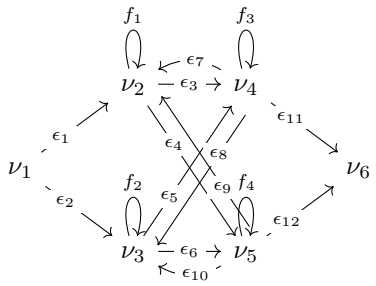

FIGURE 3. Example of Recurrent Network Quiver

and

$$\tau_{\nu_2} = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}, \quad \tau_{\nu_3} = \begin{pmatrix} 1 & 2 \\ 1 & 0 \end{pmatrix}, \quad \tau_{\nu_4} = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}, \quad \tau_{\nu_5} = \begin{pmatrix} 1 & 2 \\ 1 & -1 \end{pmatrix}.$$

Let $(U, g) = \tau(W, f)\tau^{-1}$ then (U, g) is isomorphic to (W, f). So, we obtain

$$U_{\epsilon_1} = \begin{pmatrix} 9 & 15 \\ 4 & 6 \end{pmatrix}, \quad U_{\epsilon_2} = \begin{pmatrix} 15 & 13 \\ 1 & 5 \end{pmatrix}, \quad U_{\epsilon_3} = \begin{pmatrix} 2 & 1 \\ 2 & 2 \end{pmatrix}, \quad U_{\epsilon_4} = \begin{pmatrix} 7 & 5 \\ 3 & 2 \end{pmatrix},$$

$$U_{\epsilon_5} = \begin{pmatrix} 4 & 3 \\ 4 & 1 \end{pmatrix}, \quad U_{\epsilon_6} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \quad U_{\epsilon_7} = \begin{pmatrix} 1 & 2 \\ 1 & -4 \end{pmatrix}, \quad U_{\epsilon_8} = \begin{pmatrix} 6 & -9 \\ -2 & 1 \end{pmatrix},$$

$$U_{\epsilon_9} = \begin{pmatrix} 3 & -6 \\ 0 & -1 \end{pmatrix}, \quad U_{\epsilon_{10}} = \begin{pmatrix} \frac{1}{2} & \frac{7}{2} \\ \frac{1}{5} & -\frac{5}{2} \end{pmatrix}, \quad U_{\epsilon_{11}} = \begin{pmatrix} 2 & -1 \\ -1 & 0 \end{pmatrix}, \quad U_{\epsilon_{12}} = \begin{pmatrix} 3 & 0 \\ 0 & -3 \end{pmatrix}.$$

Let us have $f_1(x) = f_2(x) = f_3(x) = f_4(x) = (x_g^2)_{g \in G}$, then we will have

$g_1(x) = \tau_{\nu_3} f_1(\tau_{\nu_3}^{-1} x), g_2(x) = \tau_{\nu_4} f_2(\tau_{\nu_4}^{-1} x), g_3(x) = \tau_{\nu_5} f_3(\tau_{\nu_5}^{-1} x), g_4(x) = \tau_{\nu_6} f_4(\tau_{\nu_6}^{-1} x)$.

We try to calculate an example of a result from recurrent neural network (W, f) and (U, g) using input $x_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ using matlab program (you can see the code https://bit.ly/3ZyrZeI) that is shown in the below.

TABLE 1. Example of a result of the neural network function

Iteration (n)	$\Psi(W, f)(x)$	$\Psi(U, g)(x)$
0	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$
1	$\begin{pmatrix} 1827103 \\ 2991981 \end{pmatrix}$	$\begin{pmatrix} 1827103 \\ 2991981 \end{pmatrix}$
2	$10^{27} \begin{pmatrix} 3.4767 \\ 6.9831 \end{pmatrix}$	$10^{27} \begin{pmatrix} 3.4767 \\ 6.9831 \end{pmatrix}$

The table shows that the neural network function of two isomorphic neural networks will be the same.

4. Moduli Space of Recurrent Neural Network

Definition 4.1. [9] *Let G be a group and X be a set. An action of G on X is a map $\cdot : G \times X \to X$, denoted by $a \cdot x$, such that*

(1) $e \cdot x = x$ for every $x \in X$ where e is the identity in G,
(2) $a \cdot (b \cdot x) = (a \cdot b) \cdot x$ for all $a, b \in G$ and all $x \in X$.

The set $\mathcal{O} = \{a \cdot x | a \in G\}$ is called an orbit of the action.

Let $\mathbb{C}^G(\mathcal{Q})$ be the set of $\mathbb{C}^G$-representations of $\mathcal{Q}$ such that for every $\epsilon \in \mathcal{E}$, W_ϵ is invertible. Define $\Gamma(\mathcal{Q})$ as a group of all isomorphisms of representations of quiver $\mathcal{Q}$. Define an action of $\Gamma(\mathcal{Q})$ on $\mathbb{C}^G(\mathcal{Q})$ as follows:

$$\tau \cdot \mathrm{W} = \tau \mathrm{W} \tau^{-1} = \{\tau_{t(\epsilon)} \mathrm{W}_\epsilon \tau_{s(\epsilon)}^{-1}\}_{\epsilon \in \mathcal{E}}. \tag{3}$$

Definition 4.2. [8] *Let $\mathcal{Q} = (\mathcal{V}, \mathcal{E}, s, t)$ be a recurrent network quiver. Let C be a subgroup of $\Gamma(Q)$. Let C act on $\mathbb{C}^G(\mathcal{Q})$ where the action is defined by $\tau \cdot \mathrm{W} = \tau \mathrm{W} \tau^{-1} = \{\tau_{t(\epsilon)} \mathrm{W}_\epsilon \tau_{s(\epsilon)}^{-1}\}_{\epsilon \in \mathcal{E}}$. The moduli space $\mathcal{M}(\mathcal{Q})$ of the $\mathbb{C}^G$-representation is the set of all orbits of the action of C on $\mathbb{C}^G(\mathcal{Q})$. The dimensions of the moduli space $\mathcal{M}(\mathcal{Q})$ is the number of orbits of the group action.*

Lemma 4.3. *Let $\mathcal{Q} = (\mathcal{V}, \mathcal{E}, s, t)$ be a cycle quiver. Let $\mu \in \mathcal{V}$ and define*

$$C_\mu(\mathcal{Q}) = \{\tau = \{\tau_\nu\}_{\nu \in \mathcal{V}} \in \Gamma(\mathcal{Q}) | \tau_\mu = 1\}.$$

Let $\mathbb{C}_\mu(\mathcal{Q})$ acts on $\mathbb{C}^G(\mathcal{Q})$ where the action is defined by $\tau \cdot \mathrm{W} = \tau \mathrm{W} \tau^{-1} = \{\tau_{t(\epsilon)} \mathrm{W}_\epsilon \tau_{s(\epsilon)}^{-1}\}_{\epsilon \in \mathcal{E}}$. The action of $C_\mu(\mathcal{Q})$ on $\mathbb{C}^G(\mathcal{Q})$ will form a moduli space $\mathcal{M}(\mathcal{Q})$ with

$$dim(\mathcal{M}(\mathcal{Q})) = |G|^2.$$

Proof. We know that $C_\mu(\mathcal{Q})$ is a subset of $\Gamma(\mathcal{Q})$. We also know that $C_\mu(\mathcal{Q})$ contains $1 = \{1_\nu\}_{\nu \in \mathcal{V}}$. Let $\tau, \sigma \in C_\mu(\mathcal{Q})$, we have $\tau \cdot \sigma^{-1} = \{\tau_\nu \sigma_\nu^{-1}\}_{\nu \in \mathcal{V}}$. If $\nu = \mu$, we have $\tau_\nu \sigma_\nu^{-1} = \tau_\mu \sigma_\mu^{-1} = 1 * 1 = 1$. If $\nu \neq \mu$, we have $\tau_\nu \sigma_\nu^{-1} \in GL(\mathbb{C}^G)$. Thus, $C_\mu(\mathcal{Q})$ is a subgroup of $\Gamma(\mathcal{Q})$. Therefore, $\mathcal{M}(\mathcal{Q})$ is a moduli space. Let $\mathcal{Q} = (\mathcal{V}, \mathcal{E}, s, t)$ be a cycle. Without losing generality, let $\mathcal{E} = \{\epsilon_0, \epsilon_1, \cdots, \epsilon_{n-1}\}$ and $\mathcal{V} = \{\nu_0, \nu_1, \cdots, \nu_{n-1}\}$ with $s(\epsilon_i) = \nu_i$ and $t(\epsilon_i) = \nu_{i+1(mod\ n)}$ where mod is the modulo operator. Let $W \in \mathbb{C}^G(\mathcal{Q})$. Now, we will construct a morphism of quiver representations $\tau = \{\tau_\nu\}_{\{\nu \in \mathcal{V}\}}$. Without losing generality, we set $\mu = \nu_0$ and for $i = 1, 2, \cdots, n-1$

$$\tau_{\nu_i} = \tau_{s(\epsilon_{i-1})} \mathrm{W}_{\epsilon_{i-1}}^{-1} = \tau_{\nu_{i-1}} \mathrm{W}_{\epsilon_{i-1}}^{-1}$$

Now, we get a new $\mathbb{C}^G$-representation of quiver Q, that is:

$$\mathrm{U} = \tau \mathrm{W} \tau^{-1} = \{\tau_{\nu_{i+1(mod\ n)}} W_{\epsilon_i} \tau_{\nu_i}\}_{i=0}^{n-1}$$

So, we get $\mathrm{U}_{\epsilon_i} = 1$ for $i = 0, 1, \cdots, n-2$ and $\mathrm{U}_{n-1} = \tau_{\nu_0} \mathrm{W}_{\epsilon_{n-1}} \tau_{\nu_{n-1}}^{-1} = \prod_{i=0}^{n-1} \mathrm{W}_{\epsilon_{n-1-i}}$. From the algorithm, we know that $W_{\epsilon_{n-1}}$ is not necessarily an identity map. Furthermore, we know that the group action is free. So the number of orbits of the action must be equivalent to the number of arrows with free weight; in this case, we have ϵ_{n-1}. On the other

hand, we know that $W_{\epsilon_{n-1}} \in GL(\mathbb{C}^G)$, where $GL(\mathbb{C}^G)$ is a general linear group of $\mathbb{C}^G$. So we can conclude that the number of orbits of the action must be the same as the size of $W_{\epsilon_{n-1}}$. Therefore, we can conclude that

$$dim(\mathcal{M}(Q) = dim(\mathbb{C}^G) = |G|^2$$

□

Theorem 4.4. *Let $\mathcal{Q}$ be a recurrent network quiver. Let $\mathcal{V}_{in}$ be the set of input vertices of Q and $\mathcal{V}_{out}$ be the set of output vertices of $\mathcal{Q}$. Define*

$$\mathcal{C}_\Gamma(\mathcal{Q}) = \{I\}_{\nu \in V_{in}} \times \prod_{v \in \tilde{V}} GL(\mathbb{C}^G) \times \{I\}_{v \in V_{out}}$$

where $GL(\mathbb{C}^G)$ is a general linear group of $\mathbb{C}^G$. Let $C_\Gamma(\mathcal{Q})$ acts on $\mathbb{C}^G(\mathcal{Q})$ where the action is defined by $\tau \cdot \mathrm{W} = \tau \mathrm{W} \tau^{-1} = \{\tau_{t(\epsilon)} \mathrm{W}_\epsilon \tau_{s(\epsilon)}^{-1}\}_{\epsilon \in \mathcal{E}}$. Let $\mathcal{M}(\mathcal{Q})$ be the set of all orbits from the action of $C_\Gamma(\mathcal{Q})$ on $\mathbb{C}^G(\mathcal{Q})$, then the set $\mathcal{M}(\tilde{\mathcal{Q}})$ will form the moduli space. Furthermore, the dimension of the moduli space is $|G|^2(|\mathcal{E}^\circ| - |\tilde{\mathcal{V}}|)$.

Proof. Firstly, we know that $C_\Gamma(\mathcal{Q})$ is a subset of $\prod_{\nu \in \mathcal{V}} GL(\mathbb{C}^G) = \Gamma(\mathcal{Q})$ and $\{1_\nu\}_{\nu \in \mathcal{V}} \in \mathbb{C}_\Gamma(\mathcal{Q})$. Let $\tau, \sigma \in \mathbb{C}_\Gamma(\mathcal{Q})$. We know that $\tau \cdot \sigma^{-1} \in \mathbb{C}_\Gamma(\mathcal{Q})$ because for $\nu \in \mathcal{V}_{in} \bigcup \mathcal{V}_{out}$, we have $\tau_\nu \sigma_\nu^{-1} = 1 * 1 = 1$ and for $\nu \in \tilde{\mathcal{V}}$, we have $\tau_\nu * \sigma_\nu^{-1} \in GL(\mathbb{C}^G)$. Thus, $C_\Gamma(\mathcal{Q})$ is a subgroup of $\Gamma(\mathcal{Q})$. Therefore, $\mathcal{M}(\mathcal{Q})$ is a moduli space. Now, we will show that the action of $\mathcal{C}_\Gamma(\tilde{\mathcal{Q}})$ on $\mathcal{M}(\tilde{\mathcal{Q}})$ by $\cdot$ is free. This means that we must show that for all $(\mathrm{W}, f) \in \mathcal{M}(\tilde{\mathcal{Q}})$ we have $\tau \cdot (\mathrm{W}, f) = (\mathrm{W}, f)$ implies $\tau = \{\tau_\nu = 1\}_{\nu \in \mathcal{V}}$. Let $(\mathcal{W}, f)$ be an element of $\mathcal{M}(\tilde{\mathcal{Q}})$. From the definition of the group action, we have

$$\tau \cdot (\mathrm{W}, f) = (\tau \cdot \mathrm{W}, \tau \cdot f) = (\{\tau_{t(\epsilon)} \mathrm{W}_\epsilon \tau_{s(\epsilon)}^{-1}\}_{\epsilon \in \mathcal{E}}, \{\tau_\nu f_\nu \tau_\nu^{-1}\}_{\nu \in \tilde{\mathcal{V}}}).$$

If $s(\epsilon)$ is in the input layer, then we have

$$\tau_{t(\epsilon)} \mathrm{W}_\epsilon \tau_{s(\epsilon)}^{-1} = \tau_{t(\epsilon)} \mathrm{W}_\epsilon$$

This means that if $\tau \cdot (\mathrm{W}, f) = (\mathrm{W}, f)$ then $\tau_{t(\epsilon)}$ must be an identity. If $s(\epsilon)$ in the first hidden layer we will also get $\tau \cdot (\mathrm{W}, f)$ implies $\tau_{t(\epsilon)} = 1$. So, using strong mathematical induction, we will get that $\tau_v = 1$ for every v in V. Thus, we have that the action of $\mathcal{C}_\Gamma(\tilde{\mathcal{Q}})$ on $\mathcal{M}(\tilde{\mathcal{Q}})$ by $\cdot$ is free.

Now, we will count arrows that have free weight. Let $\mathcal{Q}$ be a recurrent network quiver with $\tilde{\mathcal{Q}}$ as the hidden quiver and Q° as the delooped quiver. Let W be a $\mathbb{C}^G$-representation of quiver $\mathcal{Q}^\circ$. Let $\nu \in \tilde{\mathcal{V}}$. We know that there is an $\epsilon \in \mathcal{E}^\circ$ such that $t(\epsilon) = \nu$. We only choose one $\epsilon \in \mathcal{E}^\circ$ that $t(\epsilon) = \nu$ for every $\nu \in \tilde{V}$ to build a new quiver Q^ν. Because of the construction, no two arrows have the same target, which implies that Q^ν is a union of trees, and the intersection of any two trees can only be a source vertex of $\mathcal{Q}$. Furthermore, for any of those trees, only a hidden vertex is a unique source corresponding to that tree. Now, we will construct a morphism of quiver representations $\tau = \{\tau_\nu\}_{\{\nu \in \mathcal{V}\}}$. If ν is the input vertex, set $\tau_\nu = 1$. If ν is not the input vertex, we have an arrow $\alpha \in Q^\nu$ such that $t(\alpha) = \nu$. Therefore, we can set $\tau_\nu = \mathrm{W}_\alpha^{-1} \tau_{s(\alpha)}$. Using the

recursive formula, we can get a new $\mathbb{C}^G$-representation of $\mathcal{Q}^\nu$ such that every arrow in $\mathcal{Q}^\nu$ will be represented as an identity linear map from $\mathbb{C}^G$ to $\mathbb{C}^G$. We get that the number of arrows with free weight is the number of arrows in $\mathcal{Q}^\circ$ that are not necessarily the same as an identity map. On the other hand, we know that every weight of the arrow is an element of $GL(\mathbb{C}^G)$. Therefore, the dimension of the moduli space is $|G|^2(|\mathcal{E}^\circ| - |\tilde{\mathcal{V}}|)$ □

Remark 4.1. *This theorem tells us to create a new way to make a more effective and efficient recurrent neural network algorithm. We have tried to expand the approach proposed by Armenta et al.* [6],[4]. *We used group algebra to build the neural teleportation model in Theorem 4.4. Unlike the neural network with additive time-varying delays proposed by Shanmugam et al. in* [10]*, which optimizes operation between vertices, we consider some combinatorics aspects of reducing arrows between vertices.*

Example 4.5. Let $\mathcal{Q}$ be a recurrent network quiver that can be drawn as follows:

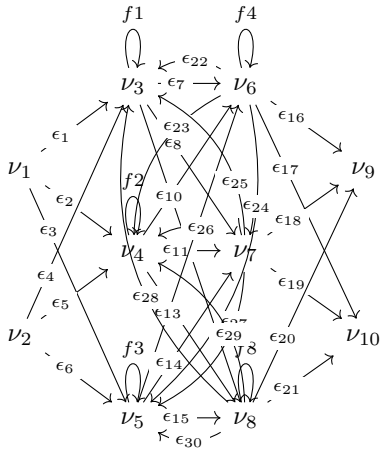

FIGURE 4. Recurrent Network Quiver $\mathcal{Q}$

We will remove all loops from the quiver and make quiver $\mathcal{Q}^\circ$.

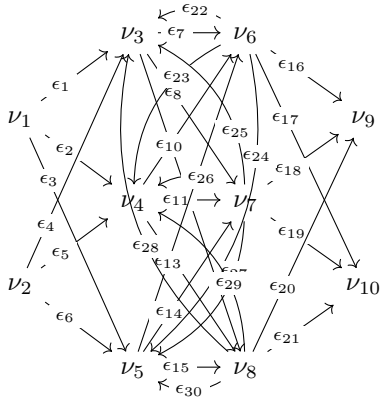

FIGURE 5. Delooped Quiver of $\mathcal{Q}$

Now, we will make a new quiver, denoted by $\mathcal{Q}^\nu$, as follows:

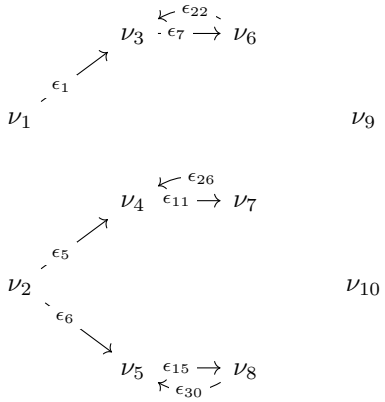

FIGURE 6. Quiver $\mathcal{Q}^\nu$

If we have (W, f) is a neural network over Q, we choose $\tau_{\nu_3} = {\mathrm{W}_{\epsilon_1}}^{-1}$, $\tau_{\nu_4} = \mathrm{W}_{\epsilon_5}^{-1}$, $\tau_{\nu_5} = \mathrm{W}_{\epsilon_6}^{-1}$, $\tau_{\nu_6} = {\mathrm{W}_{\epsilon_1}}^{-1}{\mathrm{W}_{\epsilon_7}}^{-1}$, $\tau_{\nu_7} = {\mathrm{W}_{\epsilon_5}}^{-1}{\mathrm{W}_{\epsilon_{11}}}^{-1}$, $\tau_{\nu_8} = {\mathrm{W}_{\epsilon_6}}^{-1}{\mathrm{W}_{\epsilon_{15}}}^{-1}$, and $\tau_{\nu_1} = \tau_{\nu_2} = \tau_{\nu_9} = \tau_{\nu_{10}} = 1$ Thus, we will get a new representation for $\mathcal{Q}^\nu$ as follows:
From this, we can conclude that the dimensions of the moduli spaces of recurrent neural networks over $\mathcal{Q}$ are $24 \times |G|^2$.

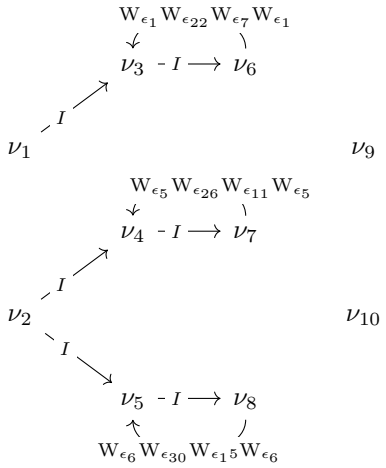

FIGURE 7. The New Representation for $\mathcal{Q}^{\nu}$

5. CONCLUSION AND FURTHER RESEARCH

We have obtained some properties of recurrent neural networks. We got the dimensions of the moduli space from the morphism group action on the set of recurrent neural networks with the invertible weight of the arrow. From this work, we can minimize the algorithm complexity of recurrent neural networks. We need help applying this model to actual data, which is possible because our model is used for raw data and is not equipped with a prepossessing data algorithm. In further research, we will try to combine topological data analysis and recurrent neural networks. We will also use the present work in future research for applications like picture recognition. We also want to see the moduli space's topology. We will combine this work with topological data analysis to create a more effective and efficient neural network algorithm.

Acknowledgement. This research was funded by an ITB Research Grant 2024.

REFERENCES

[1] Q. Zhang and Y. Wang, "Construction of composite mode of sports education professional football teaching based on sports video recognition technology," in *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, pp. 1889–1893, 2020. https://doi.ieeecomputersociety.org/10.1109/ICMCCE51767.2020.00414.

[2] F. Schöller, M. Blanke, M. Plenge-Feidenhans', and L. Nalpantidis, "Vision-based object tracking in marine environments using features from neural network detections," *IFAC-PapersOnLine (21st IFAC World Congress)*, vol. 53, no. 2, pp. 14517–14523, 2020. https://doi.org/10.1016/j.ifacol.2020.12.1455.

[3] S. Kumari, S. Aravindakshan, U. Jain, and V. Srinivasa Chakravarthy, "Convolutional elman jordan neural network for reconstruction and classification using attention window," in *Innovations in Computational Intelligence and Computer Vision* (M. K. Sharma, V. S. Dhaka, T. Perumal, N. Dey, and J. M. R. S. Tavares, eds.), (Singapore), pp. 173–181, Springer Singapore, 2021. https://doi.org/10.1007/978-981-15-6067-5_20.

[4] M. Armenta, T. Judge, N. Painchaud, Y. Skandarani, C. Lemaire, G. Gibeau Sanchez, P. Spino, and P.-M. Jodoin, "Neural teleportation," *Mathematics*, vol. 11, no. 2, 2023. https://www.mdpi.com/2227-7390/11/2/480.

[5] I. Assem, A. Skowronski, and D. Simson, *Elements of the Representation Theory of Associative Algebras: Techniques of Representation Theory*. London Mathematical Society Student Texts, Cambridge University Press, 2006.

[6] M. Armenta and P.-M. Jodoin, "The representation theory of neural networks," *Mathematics*, vol. 9, no. 24, 2021. https://www.mdpi.com/2227-7390/9/24/3216.

[7] L. C. Wanditra, I. M. Alamsyah, and G. Rachmaputri, "Wave packet transform on finite abelian group," *Southeast Asian Bulletin of Mathematics*, vol. 44, no. 6, pp. 853–857, 2020.

[8] L. C. Wanditra, I. Muchtadi-Alamsyah, and D. Nasution, "Artificial neural networks using quiver representations of finite cyclic groups," *Symmetry*, vol. 15, no. 12, 2023. https://www.mdpi.com/2073-8994/15/12/2110.

[9] I. M. Isaacs, *Algebra: a graduate course*. Graduate Studies in Mathematics 100, American Mathematical Society, 2009.

[10] S. Shanmugam, R. Vadivel, M. Rhaima, and H. Ghoudi, "Improved results on an extended dissipative analysis of neural networks with additive time-varying delays using auxiliary function-based integral inequalities," *AIMS Mathematics*, vol. 8, no. 9, pp. 21221–21245, 2023. https://doi.org/10.3934/math.20231082.

[11] M. Armenta, T. Brüstle, S. Hassoun, and M. Reineke, "Double framed moduli spaces of quiver representations," *Linear Algebra and its Applications*, vol. 650, pp. 98–131, 2022. https://doi.org/10.1016/j.laa.2022.05.018.

[12] A. Belov-Kanel, L. Rowen, and U. V. and, "Application of full quivers of representations of algebras, to polynomial identities," *Communications in Algebra*, vol. 39, no. 12, pp. 4536–4551, 2011. https://doi.org/10.1080/00927872.2011.616432.