An Introduction to Quaternion-Valued Recurrent Projection Neural Networks

Marcos Eduardo Valle
Rodolfo Anibal Lobo

University of Campinas

18 de Outubro de 2019

CAPES

UNICAMP

Objectives and Motivations

1. We are going to analyze a new recurrent neural network model based on two different architectures.

Objectives and Motivations

1. We are going to analyze a new recurrent neural network model based on two different architectures.
2. We implement associative memories to present some computational experiments.

Objectives and Motivations

1. We are going to analyze a new recurrent neural network model based on two different architectures.
2. We implement associative memories to present some computational experiments.
3. Preliminary experiments revealed that hypercomplex-valued recurrent networks, interpreted as dynamical systems, are less susceptible to chaotic behavior than their corresponding real-valued network.

Objectives and Motivations

1. We are going to analyze a new recurrent neural network model based on two different architectures.
2. We implement associative memories to present some computational experiments.
3. Preliminary experiments revealed that hypercomplex-valued recurrent networks, interpreted as dynamical systems, are less susceptible to chaotic behavior than their corresponding real-valued network.
4. Few steps to settle down at an equilibrium state.

Some Basic Concepts on Quaternions

Hypercomplex numbers that extend real and complex number systems.

$$q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \tag{1}$$

$$q = q_0 + \vec{q} \tag{2}$$

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \tag{3}$$

Application of quaternions include: Robotics, 3D Animation, Aircraft and Multi-antenna radio transmission.

Arithmetic

1. Let $q = q_0 + \vec{q}$ and $p = p_0 + \vec{p}$ be two quaternions.

$$q + p = (q_0 + p_0) + (\vec{q} + \vec{p}) \tag{4}$$

$$pq = p_0 q_0 - \vec{p} \cdot \vec{q} + p_0 \vec{q} + q_0 \vec{p} + \vec{p} \times \vec{q} \tag{5}$$

Arithmetic

1. Let $q = q_0 + \vec{q}$ and $p = p_0 + \vec{p}$ be two quaternions.

$$q + p = (q_0 + p_0) + (\vec{q} + \vec{p}) \tag{4}$$

$$pq = p_0 q_0 - \vec{p} \cdot \vec{q} + p_0 \vec{q} + q_0 \vec{p} + \vec{p} \times \vec{q} \tag{5}$$

2. The conjugate $\bar{q}$ and the norm $|q|$ are defined respectively by

$$\bar{q} = q_0 - \vec{q} \tag{6}$$

$$|q| = \sqrt{\bar{q}q} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \tag{7}$$

Arithmetic

1. Let $q = q_0 + \vec{q}$ and $p = p_0 + \vec{p}$ be two quaternions.

$$q + p = (q_0 + p_0) + (\vec{q} + \vec{p}) \tag{4}$$

$$pq = p_0 q_0 - \vec{p} \cdot \vec{q} + p_0 \vec{q} + q_0 \vec{p} + \vec{p} \times \vec{q} \tag{5}$$

2. The conjugate $\bar{q}$ and the norm $|q|$ are defined respectively by

$$\bar{q} = q_0 - \vec{q} \tag{6}$$

$$|q| = \sqrt{\bar{q}q} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \tag{7}$$

3. Unit quaternion: $|q| = 1$.

Special Function and Inner Product

1. We denote by $\mathbb{S}$ the set of all unit quaternions, i.e.,

$$\mathbb{S} = \{q \in \mathbb{H} : |q| = 1\} \tag{8}$$

Special Function and Inner Product

1. We denote by $\mathbb{S}$ the set of all unit quaternions, i.e.,

$$\mathbb{S} = \{q \in \mathbb{H} : |q| = 1\} \tag{8}$$

2. Now, we can define the quaternion-valued function $\sigma : \mathbb{H}^* \to \mathbb{S}$ given by

$$\sigma(q) = \frac{q}{|q|} \tag{9}$$

which maps the non-zero quaternions $\mathbb{H}^* = \mathbb{H} \ /\{0\}$.

Special Function and Inner Product

1. We denote by $\mathbb{S}$ the set of all unit quaternions, i.e.,

$$\mathbb{S} = \{q \in \mathbb{H} : |q| = 1\} \tag{8}$$

2. Now, we can define the quaternion-valued function $\sigma : \mathbb{H}^* \to \mathbb{S}$ given by

$$\sigma(q) = \frac{q}{|q|} \tag{9}$$

which maps the non-zero quaternions $\mathbb{H}^* = \mathbb{H} \ /\{0\}$.

3. The inner product between two quaternion-valued column vectors $x = [x_1, \ldots, x_n]^T$, $y = [y_1, \ldots, y_n]^T \in \mathbb{H}^n$ is given by

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} \bar{y}_i x_i \tag{10}$$

Hopfield Neural Networks

Hopfield Neural Networks

1. Control.

Hopfield Neural Networks

1. Control.
2. Computer Vision and Image Processing.

Hopfield Neural Networks

1. Control.
2. Computer Vision and Image Processing.
3. Classification.

Hopfield Neural Networks

1. Control.
2. Computer Vision and Image Processing.
3. Classification.
4. Optimization.

Hopfield Neural Networks

1. Control.
2. Computer Vision and Image Processing.
3. Classification.
4. Optimization.
5. Associative Memory Models.

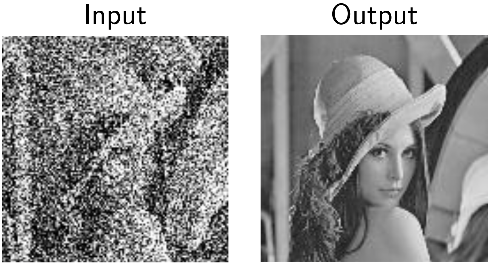
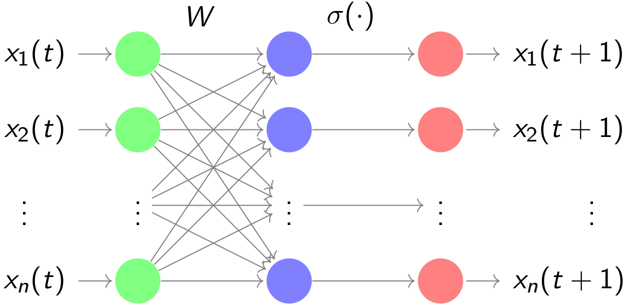


Figure: Image Recovering [Valle(2014)].

CV-QHNN Diagram Example



CV-QHNN

1. Let $w_{ij} \in \mathbb{H}$ denotes the jth quaternionic synaptic weight of the ith neuron of a network with n neurons.
2. Given $\mathbf{x}(0) = [x_1, \ldots, x_n]^T \in \mathbb{S}^n$.
3. The CV-QHNNs defines recursively the sequence of quaternion-valued vectors $\mathbf{x}(0), \mathbf{x}(1), \mathbf{x}(2), \ldots$ by

$$x_j(t+1) = \begin{cases} \sigma\left(a_j(t)\right), & 0 < |a_j(t)|, \\ x_j(t), & \text{otherwise}, \end{cases} \tag{11}$$

where

$$a_i(t) = \sum_{j=1}^{n} w_{ij} x_j(t), \tag{12}$$

Definition

1. Let $\mathcal{U} = \{\mathbf{u}^1, \ldots, \mathbf{u}^p\}$ the fundamental memory set of quaternion-valued column vectors, and with unitary components.
2. In the quaternionic version of the *correlation rule*, also called *Hebbian learning* [Isokawa et al.(2013)Isokawa, Nishimura, and Matsui], the synaptic weights are given by

$$w_{ij}^c = \frac{1}{n} \sum_{\xi=1}^{p} u_i^\xi \bar{u}_j^\xi, \quad \forall i, j \in \{1, 2, \ldots, n\}. \tag{13}$$

Definition

1. *Projection Rule*
2. Non-local storage prescription that can suppress the cross-talk effect [Kanter and Sompolinsky(1987)].
3. Formally, in the projection rule the synaptic weights are defined by

$$w_{ij}^{p} = \frac{1}{n} \sum_{\eta=1}^{p} \sum_{\xi=1}^{p} u_i^{\eta} c_{\eta\xi}^{-1} \bar{u}_j^{\xi}, \tag{14}$$

4. $$c_{\eta\xi} = \frac{1}{n} \sum_{j=1}^{n} \bar{u}_j^{\eta} u_j^{\xi} = \frac{1}{n} \left\langle \boldsymbol{u}^{\xi}, \boldsymbol{u}^{\eta} \right\rangle, \quad \forall \mu, \nu \in \{1, \ldots, p\}. \tag{15}$$

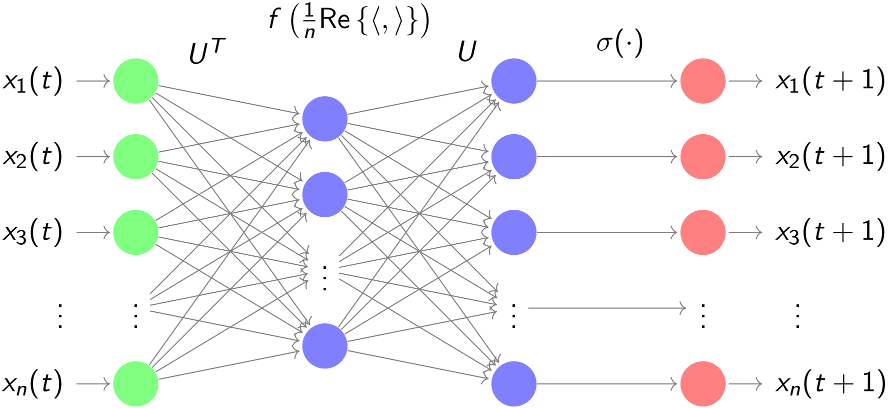
Despite its many successful applications, the Hopfield network may suffer from a very low storage capacity when is used to implement associative memories (cross-talk between the stored items).

Quaternion-Valued Recurrent Correlation Neural Networks (QRCNN)

Recurrent correlation neural networks (RCNNs), have been introduced by [Chiueh and Goodman(1991)] for the storage and recall of n-bit vectors. The RCNNs have been generalized for the storage and recall of complex-valued and quaternion-valued vectors [Valle(2014), Valle(2018)].

$$a_i(t) = \sum_{j=1}^{n} w_{ij}^{c} x_j(t) = \sum_{\xi=1}^{p} u_i^{\xi} \left[\frac{1}{n} \left\langle \mathbf{x}(t), \boldsymbol{u}^{\xi} \right\rangle \right].$$

QRCNN Diagram Example



Definition

Given a quaternionic input vector $\mathbf{x}(0) = [x_1(0), \ldots, x_N(0)]^T \in \mathbb{S}^N$, a QRCNN defines recursively a sequence $\{\mathbf{x}(t)\}_{t \geq 0}$ of quaternion-valued vectors by means of (24) where the activation potential of the ith output neuron at time t is given by

$$a_i(t) = \sum_{\xi=1}^{p} w_\xi(t) u_i^\xi, \quad \forall i = 1, \ldots, n, \tag{16}$$

with

$$w_\xi(t) = f\left(\frac{1}{n}\mathrm{Re}\left\{\left\langle \mathbf{x}(t), \boldsymbol{u}^\xi \right\rangle\right\}\right), \quad \forall \xi \in 1, \ldots, p. \tag{17}$$

Examples

1. The *correlation QRCNN* is obtained by considering in (17) the identity function $f_i(x) = x$.
2. The *high-order QRCNN*, which is determined by the function

$$f_h(x;q) = (1+x)^q, \quad q > 1. \tag{18}$$

3. The *potential-function QRCNN*, which is obtained by considering in (17) the function

$$f_p(x;L) = \frac{1}{(1-x+\varepsilon_p)^L}, \quad L \geq 1, \tag{19}$$

where $\varepsilon_p > 0$ is a small valued introduced to avoid a division by zero when $x = 1$.

4. The *exponential QRCNN*, which is determined by an exponential

$$f_e(x;\alpha) = e^{\alpha x}, \quad \alpha > 0. \tag{20}$$

Quaternion-Valued Recurrent Projection Neural Network

Quaternion-valued recurrent projection neural networks (QRPNNs) combine the main idea behind the projection rule and the QRCNN models to yield high capacity associative memories.

$$a_i(t) = \sum_{j=1}^{n} w_{ij}^{p} x_j(t) = \sum_{\xi=1}^{p} \left(\sum_{\eta=1}^{p} u_i^{\eta} c_{\eta\xi}^{-1} \right) \left[\frac{1}{n} \left\langle \mathbf{x}(t), \boldsymbol{u}^{\xi} \right\rangle \right].$$

Main Ideia

In analogy to the QRCNN, we replace the term proportional to the inner product between $\mathbf{x}(t)$ and $\boldsymbol{u}^\eta$ by the weight $w_\xi(t)$ given by (17). Accordingly, we define $c_{\eta\xi}^{-1}$ as the (η, ξ)-entry of the inverse of the real-valued matrix $C \in \mathbb{R}^{p \times p}$ given by

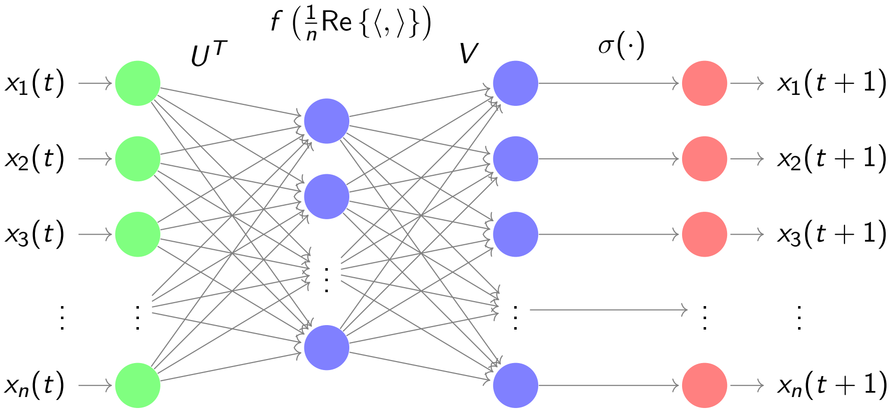
$$c_{\eta\xi} = f\left(\frac{1}{n}\mathrm{Re}\left\{\left\langle \boldsymbol{u}^\xi, \boldsymbol{u}^\eta \right\rangle\right\}\right), \quad \forall \eta, \xi \in \{1, \ldots, p\}. \tag{21}$$

Furthermore, to simplify the computation, we define

$$v_i^\xi = \sum_{\eta=1}^{p} u_i^\eta c_{\eta\xi}^{-1}, \tag{22}$$

for all $i = 1, \ldots, n$ and $\eta = 1, \ldots, p$.

QRPNN Diagram Example



QRPNNs Activation Function

Definition

The activation potential of a QRPNN is given by:

$$a_i(t) = \sum_{\xi=1}^{p} w_\xi(t) v_i^\xi, \quad \forall i = 1, \ldots, n. \tag{23}$$

where v and w are given by respectively (22) and (17) which defines recursively the sequence of quaternion-valued vectors $\mathbf{x}(0), \mathbf{x}(1), \mathbf{x}(2), \ldots$ by means of the equation

$$x_j(t+1) = \begin{cases} \sigma\left(a_j(t)\right), & 0 < |a_j(t)|, \\ x_j(t), & \text{otherwise}, \end{cases} \tag{24}$$

Example: recall probability.

1. We synthesized associative memories designed for the storage and recall of uniformly distributed fundamental memories.

Example: recall probability.

1. We synthesized associative memories designed for the storage and recall of uniformly distributed fundamental memories.
2. We probed the associative memories with an input vector $\mathbf{x}(0) = [x_1(0), \ldots, x_n(0)]^T$ obtained by replacing some components of $\mathbf{u}^1$ with probability π by an uniformly distributed component.

Example: recall probability.

1. We synthesized associative memories designed for the storage and recall of uniformly distributed fundamental memories.
2. We probed the associative memories with an input vector $\mathbf{x}(0) = [x_1(0), \ldots, x_n(0)]^T$ obtained by replacing some components of $\mathbf{u}^1$ with probability π by an uniformly distributed component.
3. The associative memories have been iterated until they reached a stationary state or completed a maximum of 1000 iterations. It succeeded if the output equals the fundamental memory $\mathbf{u}^1$.

Results

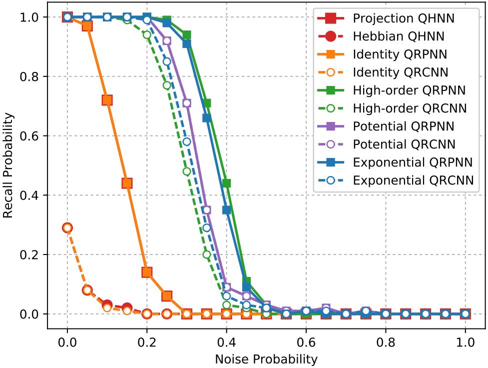


Figure: Recall Probability for Real Valued Associative Memories.

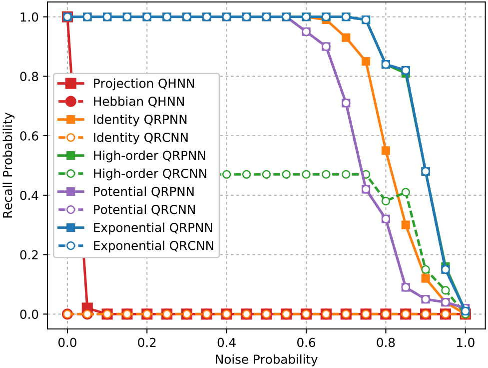
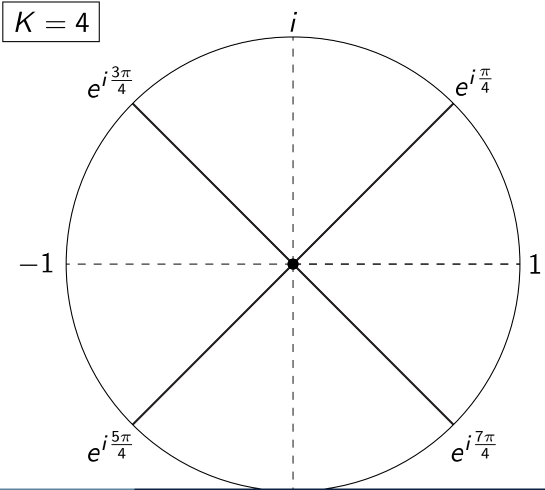


Figure: Recall Probability for Quaternion Valued Associative Memories.

Frame Title



Conclusions

1. In contrast to the QRCNNs, QRPNNs always exhibit optimal storage capacity.

Conclusions

1. In contrast to the QRCNNs, QRPNNs always exhibit optimal storage capacity.
2. The preliminary computational experiments have shown that the storage capacity and noise tolerance of QRPNNs (including real-valued case) are greater than or equal to the storage capacity and noise tolerance of their corresponding QRCNNs.

Conclusions

1. In contrast to the QRCNNs, QRPNNs always exhibit optimal storage capacity.
2. The preliminary computational experiments have shown that the storage capacity and noise tolerance of QRPNNs (including real-valued case) are greater than or equal to the storage capacity and noise tolerance of their corresponding QRCNNs.
3. We plan to investigate further the noise tolerance of the QRPNNs

Conclusions

1. In contrast to the QRCNNs, QRPNNs always exhibit optimal storage capacity.
2. The preliminary computational experiments have shown that the storage capacity and noise tolerance of QRPNNs (including real-valued case) are greater than or equal to the storage capacity and noise tolerance of their corresponding QRCNNs.
3. We plan to investigate further the noise tolerance of the QRPNNs
4. We also intend to address the performance of the new associative memories for pattern reconstruction and classification.

References I

T.D. Chiueh and R.M. Goodman.
Recurrent Correlation Associative Memories.
IEEE Trans. on Neural Networks, 2:275–284, February 1991.

Teijiro Isokawa, Haruhiko Nishimura, and Nobuyuki Matsui.
Quaternionic Neural Networks for Associative Memories.
In Akira Hirose, editor, *Complex-Valued Neural Networks*, pages 103–131. Wiley-IEEE Press, 2013.
doi: 10.1002/9781118590072.ch5.

I. Kanter and H. Sompolinsky.
Associative Recall of Memory without Errors.
Physical Review, 35:380–392, 1987.

References II

M. E. Valle.
Quaternionic Recurrent Correlation Neural Networks.
In *2018 International Joint Conference on Neural Networks (IJCNN)*, July 2018.

M.E. Valle.
Complex-Valued Recurrent Correlation Neural Networks.
IEEE Transactions on Neural Networks and Learning Systems, 25(9): 1600–1612, September 2014.
doi: 10.1109/TNNLS.2014.2341013.

THANKS!!!
OBRIGADO!!!
GRACIAS!!!