Hypercomplex-valued Neural Networks

Part 4 – Vector-Valued Extreme Learning Machines

POLISH NATIONAL AGENCY FOR ACADEMIC EXCHANGE

Marcos Eduardo Valle
Universidade Estadual de Campinas (Unicamp)
Campinas, Brazil.

Extreme Learning Machines

Extreme learning machines (ELMs) are well-established (shallow) feedforward neural networks (Huang et al., 2004).

ELMs are fully connected MLP networks in which all hidden parameters are randomly initialized and fixed.

A least-squares optimization problem performs training only on the output layer parameters.

The ELMs maintain the approximation capability while drastically decreasing the training's computational complexity.

Complex-valued and quaternion-valued ELMs have been developed by Lu et al. (2019); Lv and Zhang (2018); Minemoto et al. (2017); Zhu et al. (2021).

Vector-valued Extreme Learning Machines ($\mathbb{V}$-ELM)

Consider a single-hidden layer feedforward neural network on a finite-dimensional algebra $\mathbb{V}$.

The parameters of the single hidden layer with Q neurons are represented by a matrix $\boldsymbol{W} \in \mathbb{V}^{N \times Q}$.

Given a vector-valued row input $\mathbf{x} = [x_1, \ldots, x_N] \in \mathbb{V}^N$, the feed-forward step through the hidden layer yields

$$\boldsymbol{h} = \psi(\mathbf{x}\boldsymbol{W}) \in \mathbb{V}^Q,$$

where $\psi : \mathbb{V} \to \mathbb{V}$ is a split activation function.

The output layer parameters are arranged in a matrix $\boldsymbol{M} \in \mathbb{V}^{Q \times M}$. The output of the $\mathbb{V}$-ELM is given by

$$\mathbf{y} = \boldsymbol{h}\boldsymbol{M} \in \mathbb{V}^M.$$

Training an $\mathbb{V}$-ELM network

Consider a training set

$$\mathcal{T} = \{(\mathbf{x}_i, \boldsymbol{t}_i) : i = 1, \ldots, K\} \subset \mathbb{V}^N \times \mathbb{V}^M.$$

Organize the training elements as rows in matrices $\boldsymbol{X} \in \mathbb{V}^{K \times N}$ and $\boldsymbol{T} \in \mathbb{V}^{K \times M}$.

The hidden layer parameters are randomly generated and fixated:

$$w_{ij} = \alpha \sum_{i=1}^{n} (\texttt{randn}_i) e_i,$$

where α is a scaling factor, $\texttt{randn}_i$ yields a random number drawn from a normal distribution with mean 0 and variance 1, and $\mathcal{E} = \{e_1, \ldots, e_n\}$ is an ordered basis for $\mathbb{V}$.

The parameters of the output layer are determined by solving the vector-valued least squares problem

$$\min\{\|\boldsymbol{H}\boldsymbol{M} - \boldsymbol{T}\|_F : \boldsymbol{M} \in \mathbb{V}^{K \times Q}\} \quad \Longrightarrow \quad \boldsymbol{M} = \varphi^{-1}\left(\mathcal{M}_L(\boldsymbol{H})^\dagger \varphi(\boldsymbol{T})\right),$$

where $\boldsymbol{H} = \psi(\boldsymbol{X}\boldsymbol{W})$ is the hidden layer output matrix of the neural network.

Computational Experiments

We conducted two experiments: one featuring a time-series prediction task and one involving color image auto-encoding.

We considered the real-valued ELM and four-dimensional hypercomplex-valued ELM models.

For comparison purposes, we considered neural networks with a similar total number of trainable parameters.

A real-valued ELM with an input signal of dimension $N^{(\mathbb{R})}$, $Q^{(\mathbb{R})}$ neurons in the hidden layer, and output of dimension $M^{(\mathbb{R})}$, has

$$TNP^{(\mathbb{R})} = (Q^{(\mathbb{R})} + 1)M^{(\mathbb{R})}, \tag{1}$$

while a 4D hypercomplex-valued ELM has

$$TNP^{(\mathbb{V})} = 4(Q^{(\mathbb{V})} + 1)M^{(\mathbb{V})}. \tag{2}$$

Besides the real numbers, we considered seven 4D hypercomplex algebras:

- Cayley-Dickson algebras: $\mathbb{R}[+1,+1]$, $\mathbb{R}[+1,-1]$, $\mathbb{R}[-1,+1]$, and $\mathbb{R}[-1,-1] \simeq \mathbb{Q}$.
- Tessarines $\mathbb{T}$.
- Hyperbolic quaternions $\mathbb{Y}$.
- Klein four-group $\mathbb{K}$.

The Cayley-Dickson algebra $\mathbb{R}[-1,-1]$ corresponds to quaternions while $\mathbb{R}[-1,+1]$ is equivalent to coquaternions.

Furthermore, we have $\mathbb{R}[+1,-1] \equiv Cl(1,1)$ and $\mathbb{R}[+1,+1] \equiv Cl(2,0)$, where $Cl(p,q)$ denotes a Clifford algebra.

Times Series Prediction

For the time series prediction task, we considered the Lorenz system given by

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x), \\ \frac{dy}{dt} = x(\rho - z) - y, \\ \frac{dz}{dt} = xy - \beta z, \end{cases}$$

with $\sigma = 10$, $\beta = 8/3$, and $\rho = 28$.

A total of 4.000 consecutive positions were generated using a fourth-order Runge-Kutta method.

The first 300 positions have been used for training, while the remaining 3.700 positions have been used for testing.

We used 3 consecutive positions as input for a model to predict the next position:

- $\mathbb{R}$-ELM:
  - Inputs: $(x_{t-2}, y_{t-2}, z_{t-2}, x_{t-1}, y_{t-1}, z_{t-1}, x_t, y_t, z_t) \in \mathbb{R}^9$,
  - Output: $(x_{t+1}, y_{t+1}, z_{t+1}) \in \mathbb{R}^3$.
- $\mathbb{H}$-ELM:
  - Inputs:
    $$(\underbrace{x_{t-2}\boldsymbol{i} + y_{t-2}\boldsymbol{j} + z_{t-2}\boldsymbol{k}}_{\mathbf{p}_{t-2}}, \underbrace{x_{t-1}\boldsymbol{i} + y_{t-1}\boldsymbol{j} + z_{t-1}\boldsymbol{k}}_{\mathbf{p}_{t-1}}, \underbrace{x_t\boldsymbol{i} + y_t\boldsymbol{j} + z_t\boldsymbol{k}}_{\mathbf{p}_t}) \in \mathbb{V}^3,$$
  - Output: $\underbrace{(x_{t+1}\boldsymbol{i} + y_{t+1}\boldsymbol{j} + z_{t+1}\boldsymbol{k})}_{\mathbf{p}_{t+1}} \in \mathbb{V}$.

We evaluate the performance of the ELM models using the prediction gain (Xia et al., 2015).

We performed a series of tests with

$$Q^{(\mathbb{V})} \in \{11, 12, \ldots, 34, 35\},$$

and determined the corresponding number of hidden neurons for the real-valued ELM, resulting in

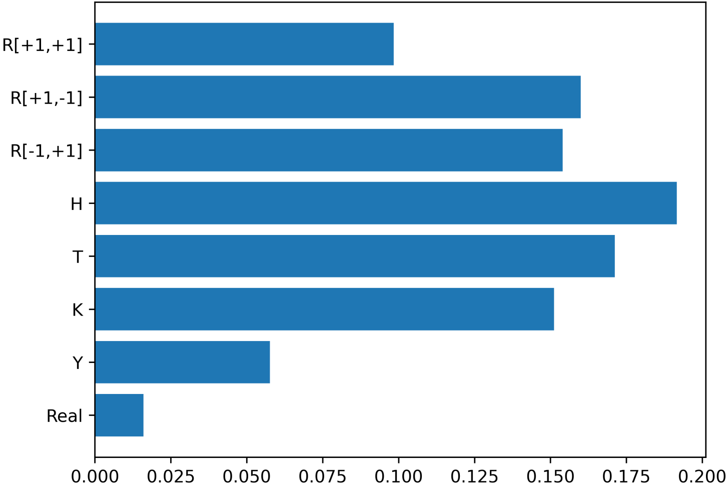
$$Q^{(\mathbb{R})} \in \{15, 16, \ldots, 45, 47\}.$$

For each $Q^{(\mathbb{V})}$ and $Q^{(\mathbb{R})}$, we trained and tested 100 networks for each algebra, resulting in a total of 20.000 simulations.

We annotated the best-performing model for each of these simulations, i.e., the model that yielded the highest prediction gain.

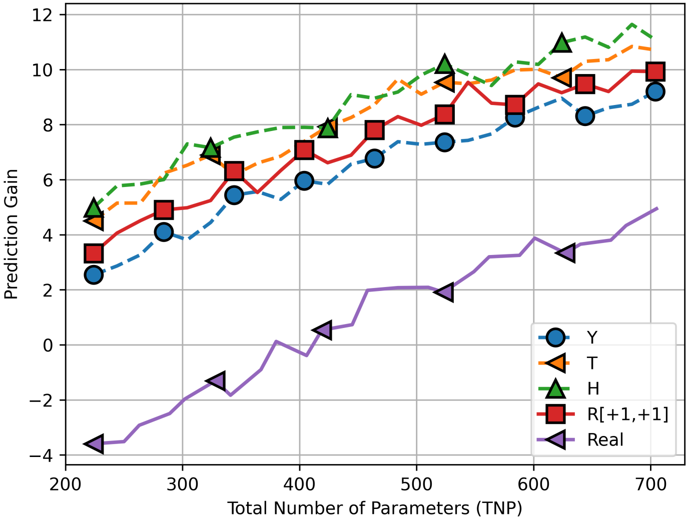
The following shows the frequency with which one model outperformed all others.

The probability of an ELM yields the highest prediction gain by the underlying algebra:



Source: Vieira and Valle (2022).

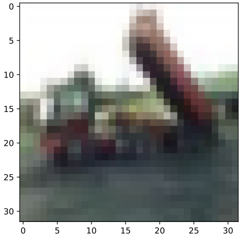
The average prediction gain by the total number of parameters:

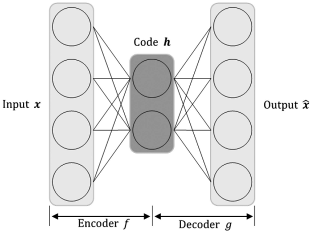


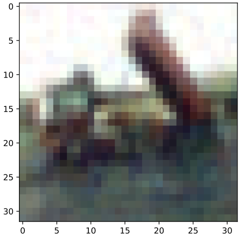
Source: Vieira and Valle (2022).

Color Image Auto-Encoding

An auto-encoder can compress a high-dimensional object and reconstruct it from the compressed information with minimal loss.







We used the CIFAR-10 dataset: $10,000$ images for train and other $10,000$ for testing.

ELM AutoEncoder

Given an input image $\mathbf{x} \in \mathbb{A}^N$ ($\mathbb{A} = \mathbb{R}$ or $\mathbb{A} = \mathbb{H}$), we have

$$\underbrace{\boldsymbol{h} = \psi(\mathbf{x}\boldsymbol{W}) \in \mathbb{A}^Q}_{\text{Enconder}} \quad \text{and} \quad \underbrace{\boldsymbol{y} = \mathbf{hM} \in \mathbb{A}^N}_{\text{Decoder}}.$$

$\mathbb{R}$-ELM: $N^{(\mathbb{R})} = 3,072$ and $Q^{(\mathbb{R})} = 600$, TNP$^{(\mathbb{R})} = 1,843,200$.

A 32×32 RGB image is converted to a real-valued vector $\mathbf{x}^{(\mathbb{R})} \in \mathbb{R}^{3072}$ concatenating the red, green, and blue channels. Also, the values are rescaled to $[-1,+1]$.

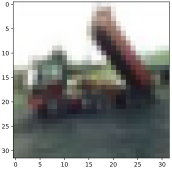
$\mathbb{V}$-ELM: $N^{(\mathbb{V})} = 1,024$ and $Q^{(\mathbb{V})} = 450$, TNP$^{(\mathbb{V})} = 1,843,200$.

A 32×32 RGB image is converted to a hypercomplex-valued vector $\mathbf{x}^{(\mathbb{V})} \in \mathbb{V}^{1024}$ by

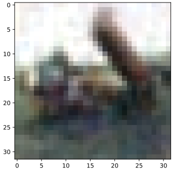
$$x_i^{(\mathbb{V})} = \left(\frac{2\mathtt{red}_i}{255} - 1\right)\boldsymbol{i} + \left(\frac{2\mathtt{green}_i}{255} - 1\right)\boldsymbol{j} + \left(\frac{2\mathtt{blue}_i}{255} - 1\right)\boldsymbol{k}.$$

training set samples: original color image and the corresponding decoded images.

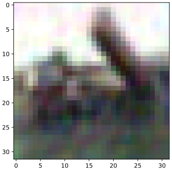
Original image:



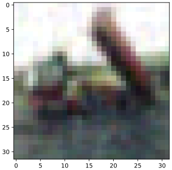
a) $\mathbb{R}$



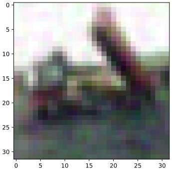
b) $\mathbb{Y}$



c) $\mathbb{K}$



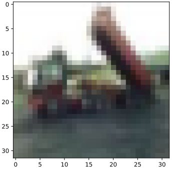
d) $\mathbb{T}$



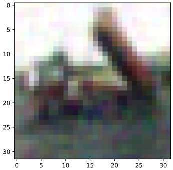
Source: Vieira and Valle (2022).

training set samples: original color image and the corresponding decoded images.

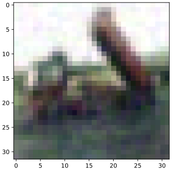
Original image:



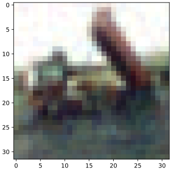
e) $\mathbb{Q}$



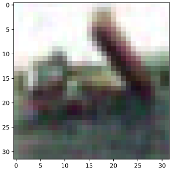
f) $\mathbb{R}[-1, +1]$



g) $\mathbb{R}[+1, -1]$



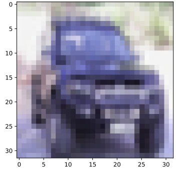
h) $\mathbb{R}[+1, +1]$



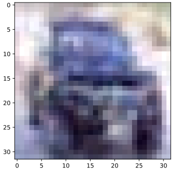
Source: Vieira and Valle (2022).

Test set samples: original color image and the corresponding decoded images.

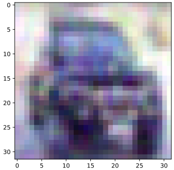
Original image:



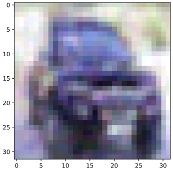
a) $\mathbb{R}$



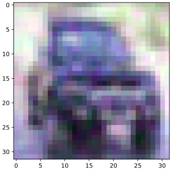
b) $\mathbb{Y}$



c) $\mathbb{K}$



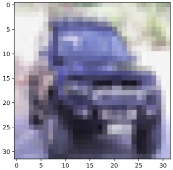
d) $\mathbb{T}$



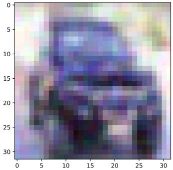
Source: Vieira and Valle (2022).

Test set samples: original color image and the corresponding decoded images.

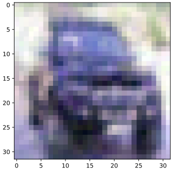
Original image:



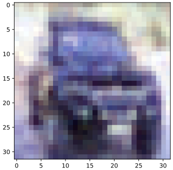
e) $\mathbb{Q}$



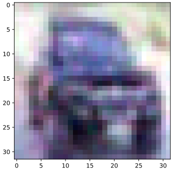
f) $\mathbb{R}[-1,+1]$



g) $\mathbb{R}[+1,-1]$



h) $\mathbb{R}[+1,+1]$



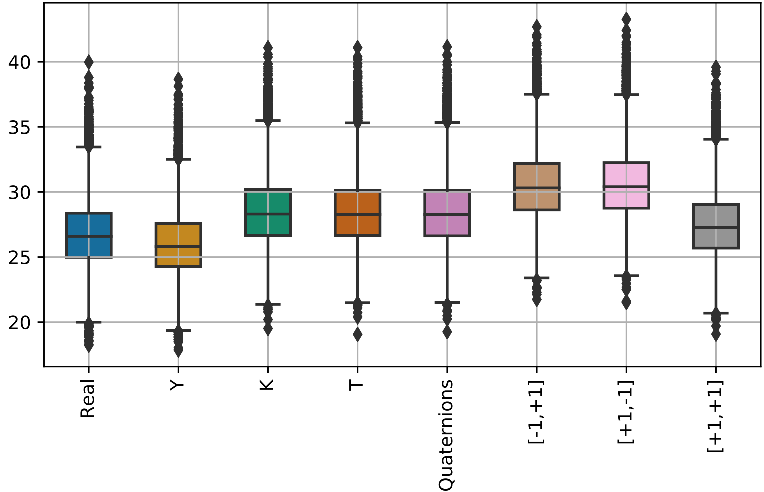
Source: Vieira and Valle (2022).

We used the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) to compare the auto-encoders.

	Train Set		Test Set	
Algebra	PSNR	SSIM	PSNR	SSIM
Real	27.3 ± 2.4	0.91 ± 0.05	26.8 ± 2.6	0.89 ± 0.05
$\mathbb{Y}$	26.4 ± 2.4	0.89 ± 0.05	26.0 ± 2.6	0.88 ± 0.05
$\mathbb{K}$	28.9 ± 2.5	0.93 ± 0.04	28.5 ± 2.7	0.92 ± 0.05
$\mathbb{T}$	28.9 ± 2.5	0.93 ± 0.04	28.5 ± 2.7	0.92 ± 0.05
$\mathbb{Q}$	28.9 ± 2.5	0.93 ± 0.04	28.5 ± 2.7	0.92 ± 0.05
$\mathbb{R}[-1,+1]$	$\mathbf{31.0} \pm \mathbf{2.5}$	$\mathbf{0.95} \pm \mathbf{0.03}$	$\mathbf{30.5} \pm \mathbf{2.7}$	$\mathbf{0.95} \pm \mathbf{0.04}$
$\mathbb{R}[+1,-1]$	$\mathbf{31.1} \pm \mathbf{2.5}$	$\mathbf{0.95} \pm \mathbf{0.03}$	$\mathbf{30.6} \pm \mathbf{2.7}$	$\mathbf{0.95} \pm \mathbf{0.04}$
$\mathbb{R}[+1,+1]$	27.9 ± 2.4	0.92 ± 0.04	27.5 ± 2.6	0.91 ± 0.05

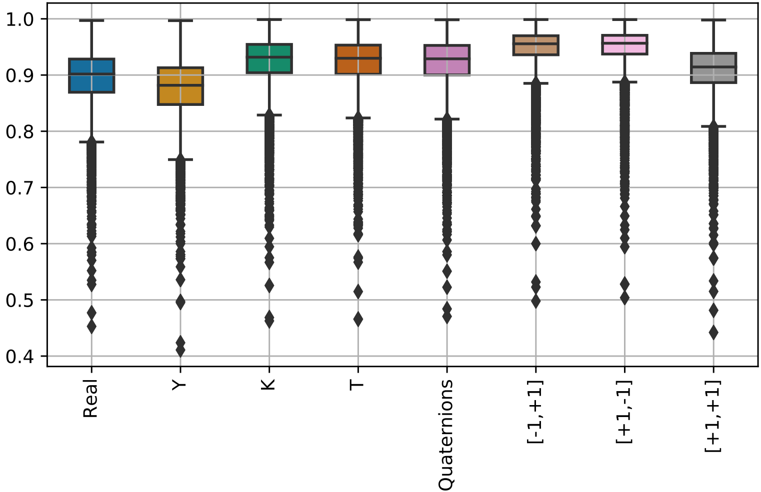
Average PSNR and SSIM rates.

PSNR rates:



Source: Vieira and Valle (2022).

SSIM rates:



Source: Vieira and Valle (2022).

Concluding Remarks

This talk addressed vector-valued extreme learning machines.

We implemented seven four-dimensional hypercomplex-valued ELM models besides the traditional real- valued ELM.

The neural networks have been used for chaotic time series prediction and an auto-encoding task.

The hypercomplex-valued models outperformed the traditional real-valued ELM by a noticeable margin on both tasks.

Less known algebras, such as the Cayley-Dickson algebras, may perform better in some machine learning tasks.

Thanks for your attention!

Acknowledge

These slides are part of a mini-course given during the workshop on **hypercomplex-valued neural networks**, which took place at the **Institute for Research and Applications of Fuzzy Modeling**, **University of Ostrava**, Ostrava, Checz Republic, *06-10 February 2023*, with the support of

NAWA POLISH NATIONAL AGENCY FOR ACADEMIC EXCHANGE

My research on hypercomplex-valued neural networks has been partially supported by:

FAPESP and CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico

References (1)

G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, and others. Extreme learning machine: a new learning scheme of feedforward neural networks. *Neural networks*, 2:985–990, 2004.

L. Lu, X. Zhang, and X. Xu. Hypercomplex extreme learning machine with its application in multispectral palmprint recognition. *PLOS ONE*, 14(4):e0209083, 4 2019. ISSN 1932-6203. doi: 10.1371/JOURNAL.PONE.0209083. URL `https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0209083`.

H. Lv and H. Zhang. Quaternion Extreme Learning Machine. In *Proceedings of ELM-2016*, pages 27–36. Springer, 2018.

T. Minemoto, T. Isokawa, H. Nishimura, and N. Matsui. Feed forward neural network with random quaternionic neurons. *Signal Processing*, 136:59–68, 2017. doi: 10.1016/j.sigpro.2016.11.008.

References (2)

G. Vieira and M. E. Valle. A general framework for hypercomplex-valued extreme learning machines. *Journal of Computational Mathematics and Data Science*, 3:100032, 6 2022. ISSN 2772-4158. doi: 10.1016/J.JCMDS.2022.100032. URL https://linkinghub.elsevier.com/retrieve/pii/S2772415822000062.

Y. Xia, C. Jahanchahi, and D. P. Mandic. Quaternion-Valued Echo State Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 26:663–673, 2015.

References (3)

S. Zhu, H. Wang, H. Lv, and H. Zhang. Augmented Online Sequential Quaternion Extreme Learning Machine. *Neural Processing Letters 2021 53:2*, 53(2):1161–1186, 2 2021. ISSN 1573-773X. doi: 10.1007/S11063-021-10435-8. URL https://link.springer.com/article/10.1007/s11063-021-10435-8.