A Hyperbolic Multilayer Perceptron

Sven Buchholz Gerald Sommer

Department of Computer Science, University of Kiel
Preusserstr. 1-9, 24105 Kiel, Germany
{sbh,gs}@ks.informatik.uni-kiel.de

Abstract

In this paper we present a novel MLP–type neural network based on hyperbolic numbers — the Hyperbolic Multilayer Perceptron (HMLP). The neurons of the HMLP compute 2D–hyperbolic orthogonal transformations as weight propagation functions. The HMLP can therefore be seen as the hyperbolic counterpart of the known Complex MLP. The HMLP is proven to be a universal approximator. Furthermore, a suitable Backpropagation algorithm for it is derived. It is shown by experiments, that the HMLP can learn tasks with underlying hyperbolic properties much more accurately and efficiently than a Complex MLP and an ordinary MLP.

1 Introduction

A growing interest in neural networks in non–Euclidean spaces can be observed in the literature, in particular in hyperbolic ones. However, most of the approaches use actually a embedding in a Euclidean space. Examples are [7] for a embedding of the hyperbolic plane and [4] for only reconstructing hyperbolic metrics. This requires always an expanding preprocessing of the data, which makes such approaches less flexible. In this paper we present instead a novel MLP-type neural network that acts directly in the hyperbolic number algebra — the Hyperbolic Multilayer Perceptron (HMLP). The neurons of the HMLP compute 2D–hyperbolic orthogonal transformations as weight propagation functions. The HMLP can therefore be seen as the hyperbolic counterpart of the known Complex MLP.

The paper starts with a preliminary section on hyperbolic numbers. In the main part of this paper, section 3, the architecture of the HMLP is presented and the HMLP is proven to be a universal approximator. Furthermore, a suitable Backpropagation algorithm for it is derived there and it is argued that the derived learning algorithm is robust, although performed in a domain containing divisors of zero. In section 4 experiments are reported, that show that the HMLP can learn tasks with underlying hyperbolic properties much more accurately and efficiently than a Complex MLP and an ordinary MLP.

2 Hyperbolic Numbers

Hyperbolic (or double) numbers are numbers of the form

$$\boldsymbol{h} = a + b\,\mathrm{e} \tag{1}$$

with $a, b \in \mathbb{R}$ and an imaginary unit e that squares to $+1$. In complete analogy with the complex numbers, a is called the real part of $\boldsymbol{h}$ and $b\,\mathrm{e}$ the imaginary part of $\boldsymbol{h}$. The conjugate hyperbolic number to $\boldsymbol{h}$ is given by $\bar{\boldsymbol{h}} := a - b\,\mathrm{e}$. The square root of the real number $|\boldsymbol{h}\,\bar{\boldsymbol{h}}| = |a^2 - b^2|$, whose sign agrees with the sign of the larger of a and b in absolute value, is called the modulus of $\boldsymbol{h}$ and is denoted by $|\boldsymbol{h}|$. Formal multiplication subject to $e^2 = 1$ gives the following definition of the product of two hyperbolic numbers

$$(a + b\,\mathrm{e}) \otimes (c + d\,\mathrm{e}) := (ac + bd) + (ad + bc)\,\mathrm{e}\,. \tag{2}$$

However, division

$$\frac{\boldsymbol{k}}{\boldsymbol{h}} = \frac{c + d\,\mathrm{e}}{a + b\,\mathrm{e}} = \frac{ca - db}{a^2 - b^2} + \frac{da - cb}{a^2 - b^2}\,\mathrm{e} \tag{3}$$

is possible only if $|\boldsymbol{h}| \neq 0$. From (2) follows that $\mathbb{H} := (\mathbb{R}^2, +, \otimes)$ is a real associative and commutative algebra. Yet, since (3) it is not a division algebra. The hyperbolic number algebra $\mathbb{H}$ is isomorphic to the matrix algebra generated by $\left\{ \begin{pmatrix} a & b \\ b & a \end{pmatrix} \mid a,b \in \mathbb{R} \right\}$ [8]. Let us point out now in what sense $\mathbb{H}$ is indeed hyperbolic. Any hyperbolic number $\boldsymbol{h}$ with non–zero modulus r can be written as $\boldsymbol{h} = a + b\,\mathrm{e} = r(\frac{a}{r} + \frac{b}{r}\,\mathrm{e})$. Hence it follows that $\frac{a}{r} = \cosh(\phi), \frac{b}{r} = \sinh(\phi)$ or $\frac{a}{r} = \sinh(\phi), \frac{b}{r} = \cosh(\phi)$. From this we get the representation $\boldsymbol{h} = r(\cosh(\phi) + \mathrm{e}\sinh(\phi))$ or $\boldsymbol{h} = r(\sinh(\phi) + \mathrm{e}\cosh(\phi))$. Thus, multiplication of a hyperbolic number with non–zero modulus is a scaled 2D–hyperbolic orthogonal transformation. More precisely,

$$(\{\boldsymbol{u} \in \mathbb{H} \mid : |\boldsymbol{u}| \pm 1\}, \otimes) \simeq O(1,1) = \left\{ \begin{pmatrix} \epsilon_1 \cosh(\phi) & \epsilon_2\, sinh(\phi) \\ \epsilon_1 \sinh(\phi) & \epsilon_2\, cosh(\phi) \end{pmatrix} \mid \phi \in \mathbb{R}, \epsilon_1, \epsilon_2 \in \{\pm 1\} \right\}. \quad (4)$$

3 Hyperbolic MLP

From the previous section we know already that hyperbolic numbers are the hyperbolic counterpart of complex numbers. In this section we will develop now the corresponding Hyperbolic Multilayer Perceptron (HMLP), which can be seen as the counterpart of the Complex MLP (CMLP) [1],[5] again.

3.1 HMLP Architecture

The HMLP as a MLP–type neural network consists of layers of neurons (say L) with feed–forward only connections between all the neurons of consecutive layers. In contrast to the real–valued MLP all entities are now hyperbolic numbers. More precisely, the output $\boldsymbol{o_j}^{(l)}$ of the j-th neuron in layer $1 < l \leq L$ of a HMLP is given by

$$\boldsymbol{o_j}^{(l)} = \boldsymbol{f}((\sum_i \boldsymbol{w_{ij}}^{(l)} \otimes \boldsymbol{x_i}^{(l-1)}) + \boldsymbol{\theta_j}^{(l)}), \quad (5)$$

where $\boldsymbol{w_{ij}^{(l)}}$ is the weight connecting the i–th node in layer $(l-1)$ with the j–th node in layer l, $\boldsymbol{\theta_j}$ is the appropriate bias and $\boldsymbol{x_i^{(l-1)}}$ is the i-th input from the previous layer. Thus, the HMLP performs as weight association a scaled 2D–hyperbolic orthogonal transformation, instead of the scalar product (MLP) and a scaled 2D–Euclidean orthogonal transformation (CMLP), respectively. Figure 1 gives an illustration of the decision regions of a linear ($\boldsymbol{f} = id$) hyperbolic neuron, which are always defined by the asymptotes of a hyperbola.

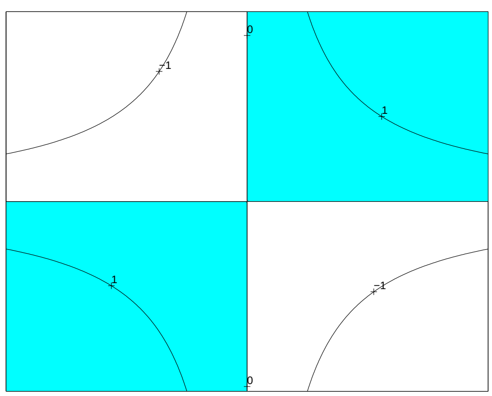

Figure 1: Decision regions of a linear hyperbolic neuron

Note that $\boldsymbol{\theta}$ acts as a translation parameter and without normed weights the defined hyperbola is arbitrarily, and so will be its two intersecting asymptotes.
In the complex case it is common [1] to use the real logistic function $\sigma : x \mapsto 1/(1 + \exp(-x))$ in each component, to avoid technical problems [5]. For similar reasons we will also use this function. Hence, the activation function of HMLP–neuron is given by

$$\boldsymbol{\sigma}(\boldsymbol{h}) = \sigma(a) + \sigma(b)\,\mathrm{e}\,. \quad (6)$$

A density theorem for HMLPs with this activation function is easy to prove.

Theorem 1 *Let X be a compact subset of $\mathbb{H}^n$. Then there exists a natural number N such that the space*

$$\left\{ \sum_{j=1}^{N} \lambda_j \, \boldsymbol{\sigma}((\sum_{i=1}^{n} \boldsymbol{w_i} \otimes \boldsymbol{x_i}) + \boldsymbol{\theta_j}) \right\} \tag{7}$$

is dense in the space of all continuous functions from X to $\mathbb{H}$.

Proof A set of $\mathbb{H}$–valued functions has the universal approximation property, iff it is a universal approximator for any of the (real–valued) component functions. However, this is already guaranteed by the fundamental density theorem for MLPs with sigmoidal functions [3]. □

3.2 Hyperbolic MLP Learning Algorithm

In this section we want to derive a Backpropagation algorithm for a HMLP with L layers. This can be done by applying gradient descent to minimize the common error function

$$E = \frac{1}{2} \sum_p (\boldsymbol{y_p} - \boldsymbol{o_p}^{(L)})^2 \,, \tag{8}$$

whereby $\boldsymbol{y_p}$ denotes the p-th expected output value. In addition to the notations used so far let $\boldsymbol{a_j}^{(l)} := (\sum_i \boldsymbol{w_{ij}}^{(l)} \otimes \boldsymbol{x_i}^{(l-1)}) + \boldsymbol{\theta_j}^{(l)}$ be the activation value of the j–th neuron in layer l ($l > 1$). Then the formula for updating the weights of the output layer is given by

$$\Delta \boldsymbol{w_{ij}}^{(L)} = [\, \underbrace{(\boldsymbol{y_p} - \boldsymbol{o_p}^{(L)}) \odot \dot{\boldsymbol{\sigma}}(\boldsymbol{a_j}^{(L)})}_{\boldsymbol{\delta_j}^{(L)}} \,] \otimes \boldsymbol{x_i}^{(L-1)} \,, \tag{9}$$

with $\odot$ denoting scalar multiplication component by component. The rule to update the hidden weights ($1 < l < L$) is as follows

$$\Delta \boldsymbol{w_{ij}}^{(l)} = [\, (\sum_k \boldsymbol{w_{jk}}^{(l+1)} \otimes \boldsymbol{\delta_k}^{(l+1)}) \odot \dot{\boldsymbol{\sigma}}(\boldsymbol{a_j}^{(l)}) \,] \otimes \boldsymbol{x_i}^{(l-1)} \,. \tag{10}$$

Finally, the bias updating is performed according to $\Delta \boldsymbol{\theta_j}^{(l)} = \boldsymbol{\delta_j}^{(l)}$. Since $\mathbb{H}$ contains divisors of zero, it is not always guaranteed that a non–zero error $\boldsymbol{\delta_j}^{(l)}$ results in a changing of weights. In [1] it is claimed that in such cases learning stops. This is not true in general, since such a effect in practice will only be temporary due to the cycling trough the remaining input data with changing the state of the network. Let us have in addition a direct look at zero divisors in our hyperbolic case. They have the special form $(a + a\,\mathrm{e}) \otimes (b - b\,\mathrm{e})$, from which we can conclude that already their appearance is unlikely. For a more general and complete discussion see [2].

4 Experiments

The HMLP can be applied to any kind of function approximation task, since we know from section 3 that it is a universal approximator. However, it should clearly be most useful on problems with underlying hyperbolic geometric reasoning. Note that it is not easy to predict if a CMLP or a MLP performs better on a given task, even if the data consist of complex numbers [6]. To the test the performance of the HMLP we considered the task to approximate a slightly hyperbolic deformed sphere given by the equations ($\phi \in (-\pi, \pi)$, $\psi \in (-\frac{\pi}{2}, \frac{\pi}{2})$)

$$x(\phi, \psi) = \cos(\phi) cosh(\psi)/12 \tag{11}$$

$$y(\phi, \psi) = \cos(\phi) sin(\psi) \tag{12}$$

$$z(\phi, \psi) = \sin(\phi) \,. \tag{13}$$

Figure 2 shows this surface generated from a uniform 16×16 grid in $[0,1] \times [0,1]$, which also served as test data in the simulations.

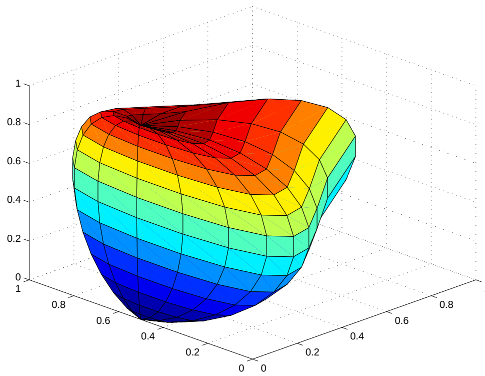

Figure 2: Test data for the simulations

The training set used consisted of 20 points randomly drawn from $[0,1] \times [0,1]$ with uniform probability. The output for both the HMLP and the CMLP was coded according to $\{(x,y),(z,0)\}$. A satisfying generalization result could be obtained by a HMLP with only 3 hidden nodes, see Figure 3 below.

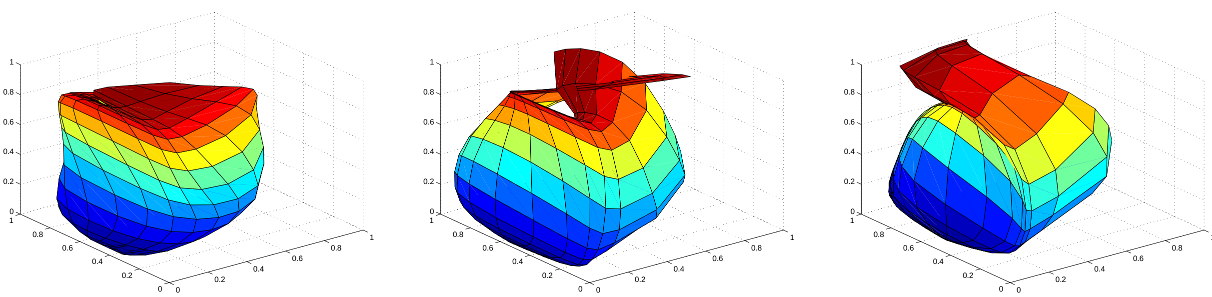

Figure 3: Generalization of the (1-3-2)–HMLP (left), the (1-3-2)–CMLP (middle) and the (2-4-3)–MLP (right)

In contrast to that, neither a CMLP nor a MLP with the same degrees of freedom (number of weights counted as real numbers) could achieved such a good performance. Obviously from Figure 3, both failed to detect a closed surface. The first 3 columns of Table 1 report the obtained results over 10 averaged trials. On the test set the mean–square–error (MSE) of the HMLP was only a half of that of the other two networks. This is particular remarkable in the case of the CMLP that achieved training errors not far above that of the HMLP.

	(1-3-2)–CMLP	(2-4-3)–MLP	(1-3-2)–HMLP	(1-4-2)–CMLP	(2-7-3)–MLP
MSE Training	0.0051	0.0082	0.0047	0.0013	0.0017
MSE Test	0.0340	0.0372	0.0176	0.0244	0.0209

Table 1: Summary of simulation results

We then increased the number of hidden nodes in the CMLP and MLP, respectively, until the training errors dropped under that of the HMLP for the first time. The parameters and the obtained results can be taken from the last two columns of Table 1. As listed there, the error on the test set is still somewhat higher than that of the HMLP besides much smaller training errors. Both networks still suffer on missing the right model of the data, which can be seen from Figure 4.

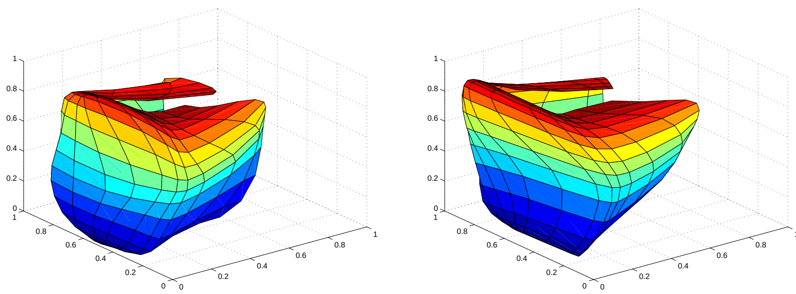

Figure 4: Generalization of the (1-4-2)–CMLP (left) and the (2-7-3)–MLP (right)

Summarizing, we can conclude that the HMLP outperformed the other two network types on the given task.

5 Conclusion

We presented a novel MLP–type neural network — the Hyperbolic Multilayer Perceptron (HMLP), that can be seen as the counterpart of the known Complex MLP. The HMLP was proven to be a universal approximator performing 2D–hyperbolic orthogonal transformations as weight propagation function. From the experiments that were made, it can be concluded that the HMLP can learn tasks with underlying hyperbolic properties much more accurately and efficiently than a Complex MLP and an ordinary MLP. However, the data used in the simulations was only slightly hyperbolic modeled. Thus, the HMLP should perform even better on pure hyperbolic tasks in comparison to other MLPs. Moreover, it seems also be promising to test the HMLP on other type of tasks and standard benchmarks.
This will be the subject of ongoing future work.

Acknowledgment This work was supported by the Deutsche Forschungsgemeinschaft under grant So-320-2-1.

References

[1] P. Arena, L. Fortuna, G. Muscato, and M. G. Xibilia. *Neural Networks in Multidimensional Domains.* Number 234 in LNCIS. Springer–Verlag, 1998.

[2] S. Buchholz and G. Sommer. *Introduction to Neural Computation in Clifford Algebra*, chapter 13. G. Sommer (Ed.), Geometric Computing with Clifford Algebra. Springer–Verlag, 2000 (to appear).

[3] G. Cybenko. Approximation by superposition of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

[4] W. Duch, R. Adamczak, and G.H.F. Diercksen. Neural Networks In Non–Euclidean Spaces. *Neural Processing Letters*, 10(3):201–210, 1999.

[5] G. Georgiou and C. Koutsougeras. Complex domain backpropagation. *IEEE Trans. on Circuits and Systems II*, 39(5):330–334, 1992.

[6] T. Masters. *Signal and Image Processing with Neural Networks.* John Willey and Sons, 1994.

[7] H. Ritter. *Self–Organizing Maps in non–euclidean Spaces*, chapter 7. E. Oja and S. Kaski (Eds.), Kohonen Maps. Elsevier, 1999.

[8] V. G. Shervatov. *Hyperbolic Functions.* D.C. Heath and Company, Boston, 1963.