Preprint of an article published in Applied Optics, vol.35, no.26, September 1996.

Incorporating LCLV Non-Linearities in Optical Multilayer Neural Networks

P. Moerland, E. Fiesler, and I. Saxena

Abstract: Sigmoid-like activation functions as available in analog hardware differ in various ways from the standard sigmoidal function as they are usually asymmetric, truncated, and have a non-standard gain. We present an adaptation of the backpropagation learning rule to compensate for these non-standard sigmoids. This method is applied to multilayer neural networks with all-optical forward propagation and liquid crystal light valves (LCLV) as optical thresholding devices. In this paper, the results of simulations of a backpropagation neural network with five different LCLV response curves as activation functions are presented. While performing poorly with the standard backpropagation algorithm, it is shown that our adapted learning rule performs well with these LCLV curves.

Keywords: *(artificial) neural network, optical multilayer neural network, hardware implementation, liquid crystal light valve (LCLV), activation function, curve fit, gain.*

1 Introduction

Optical implementations of multilayer neural networks (NNs) are promising for a variety of reasons. First of all, light offers the fastest possible communication channel, not requiring physically limiting conductors. Secondly, intersecting light beams do not noticeably interfere with each other. This means that the large number of interconnections of a NN can be optically implemented in a compact way, paving the way for truly parallel implementations of large NNs. Most optical implementations of multilayer NNs perform non-linear thresholding, which is an essential constituent of all NN models, electronically [1] [2], and hence involve conversion of optical signals to electronic ones and vice versa. In order to avoid this conversion and to progress to all-optical forward propagation in multilayer NNs, the use of optical activation functions is essential. Foremost is the use of liquid crystal light valves as non-linear optical activation functions [3] [4] [5], since their response curves are sigmoid-like. Common to most of these approaches is the fact that a thorough mathematical description of the optical activation functions and an analysis of their differences with ideal thresholding functions is lacking. Furthermore, the effects of using these non-standard activation functions in training a multilayer NN are effectively not evaluated.

In this paper, five different sigmoid-like activation functions realized by LCLVs, to be referred to as LCLV activation functions in the rest of this paper, are described and their usefulness evaluated. To be able to characterize various properties of these sampled activation functions, which are described by experimental *response data*, an approximation by a generic sigmoid *curve fit* is given. These LCLV activation functions differ, like many activation functions implemented in hardware, from the standard sigmoid since they are truncated, translated along the x-axis, and have a non-standard gain. A wide range of simulations with the backpropagation learning rule has been performed, both with the sigmoid curve fits of the LCLV response curves and with the LCLV response data. Since the standard backpropagation algorithm usually fails to converge when using these LCLV activation functions, the backpropagation algorithm has been adapted to compensate for differences between non-standard

sigmoids realized in hardware, like the LCLV activation functions, and the standard sigmoid. The performance shown by this adapted learning rule with all LCLV activation functions demonstrates its wide ranging applicability in an adaptive multilayer optical neural network.

Desirable features for the field of optical neural networks are the ability of performing subtraction and an ideal optical non-linearity which is spatially uniform in its response. In general, spatial non-uniformities measured as a variation in the read-out intensities, are present in LCLVs. This can limit the usable area of the LCLV surface to a part over which the variation is acceptably small. In addition, the fan-out optics and other optical elements may have non-ideal behaviour. These non-uniformities are expected to be compensated for to a considerable extent in an adaptive optical neural network [6], the weights of which are updated during the training on the actual optical system. If required, an additional mask could also be inserted into the optical system having a spatially varying transmission which complements (and thereby compensates for) existing spatial non-uniformities [5].

The lack of a mechanism for intensity subtraction is a limiting factor for the use of optical neural networks. Using polarization modulating devices like LCLVs, image subtraction can be performed [7] independently in a stage previous to that of thresholding. It would, however, be very desirable to be able to do this in the same processing plane (of the LCLV) as for thresholding. Techniques which facilitate implementation of subtraction using existing LC devices would significantly contribute to the development of MONNs. In this paper the focus is on the use of LCLVs as activation functions in a multilayer network. Therefore, the possibility of negative weights has been allowed in the matrix-vector multiplier (MVM). The resulting MVM output values are truncated at the origin to obtain all positive inputs to the LCLV thresholding function, such that a subtraction at the LCLV is not required as successful simulation performance shows.

The structure of the rest of this paper is as follows. In section 2 a description is given of an adaptive multilayer neural network that enables all-optical forward propagation and in which the non-linearities are realized by LCLVs. Next, the five LCLV activation functions are discussed based on their approximation by a generic sigmoid fit. In section 4 the benchmarks employed in the simulations are reported on. First, the influence of the translation of the LCLV activation functions along the $x-$axis is examined and compensated for. Next, the effect of their non-standard gain is outlined and it is reported how this effect can be counterbalanced by a rigorous mathematical solution.

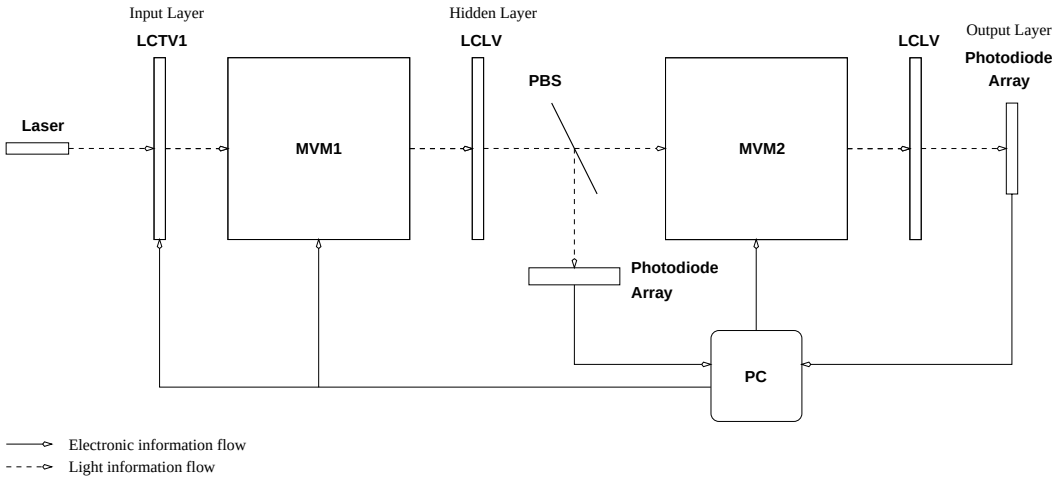

Figure 1: Schematic of the Optical Multilayer Neural Network.

2 Adaptive Optical Multilayer Neural Network Description

Saxena and Fiesler [8] have described an adaptive multilayer optical NN with a large number of interconnections and all-optical forward propagation. This three layer neural network (a layer of a neural network is defined as a layer of neurons, see [9]) will be realized as a neurocomputer consisting

of optical hardware, a computer, and an interface between them (figure 1). The optical system uses liquid crystal televisions (LCTVs) to implement adaptive interconnection weight matrices permitting learning and LCLVs to implement non-linear thresholding. Each of the matrix-vector multipliers (MVM) in figure 1 takes as input a pattern, for example a handwritten character, represented as a 2-D image , that is replicated on an LCTV representing the weight matrix, followed by an integrating micro-lens array which gives the MVM output vector. The information flow through this network during training can be described as follows. First, an input pattern is presented to the input layer, represented by a liquid crystal television screen (LCTV1). Next, the weights of the interconnects are presented to the LCTVs in MVM1 and MVM2, respectively, by the computer. The matrix-vector product of inputs and weights is calculated (MVM1) and the outcome is thresholded by the LCLV to form the outputs of the hidden layer. These hidden layer outputs are similarly presented to MVM2 to calculate the second matrix-vector product (MVM2), while thresholding of this final product is done by the LCLV. Photodiode arrays are used to capture the resulting output and hidden layer activation values which are input to the computer to calculate the next set of weight updates. This process is repeated until the network has been trained. It is in the context of such an adaptive optical multilayer network that the LCLV functions which are the subject of the rest of this paper are envisaged to be employed.

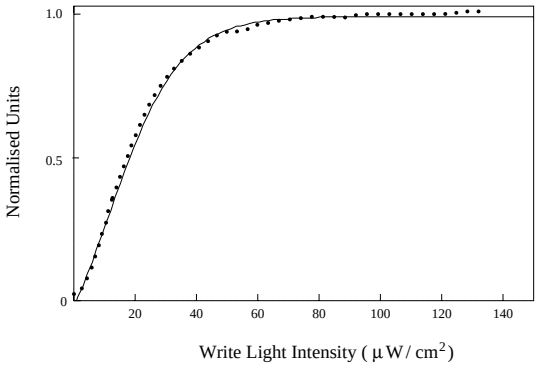

Figure 2: Response curve of LCLV1 (dotted line) and its curve fit (solid line).

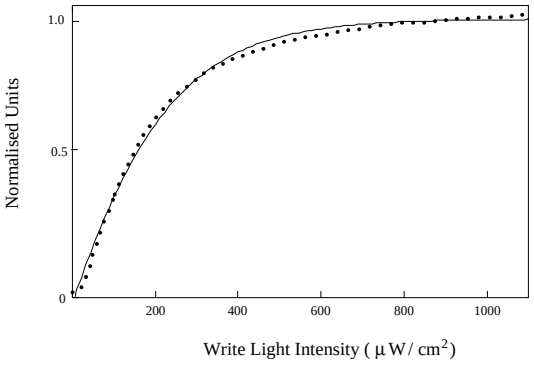

Figure 3: Response curve of LCLV2 (dotted line) and its curve fit (solid line).

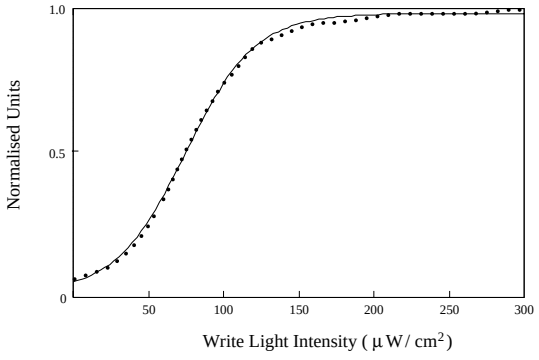

Figure 4: Response curve of LCLV3 (dotted line) and its curve fit (solid line).

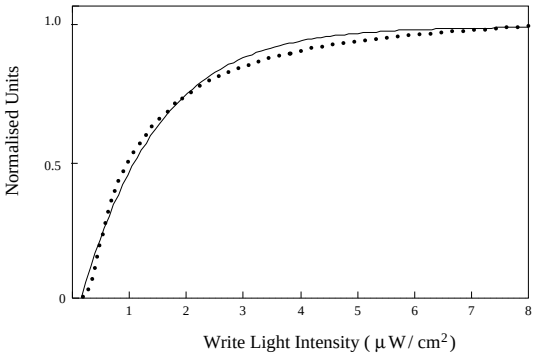

Figure 5: Response curve of LCLV4a (dotted line) and its curve fit (solid line).

3 Optical Activation Functions

A common choice for the non-linear activation function of a neuron in a multilayer neural network is the logistic or standard sigmoid function $1/(1+e^{-x})$. However, the optical activation functions realized

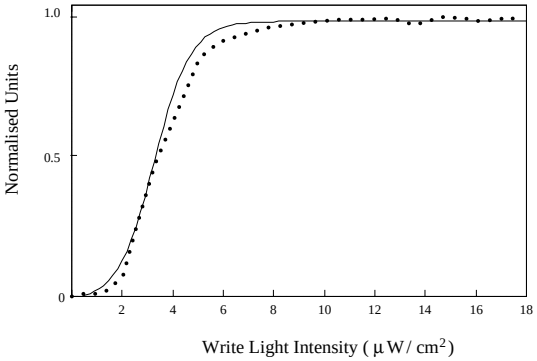

Figure 6: Response curve of LCLV4b (dotted line) and its curve fit (solid line).

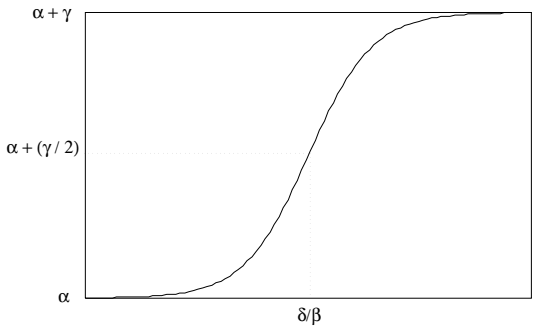

Figure 7: Generic sigmoid function.

by LCLVs differ in several ways from this standard sigmoid. One of the differentiating characteristics of LCLV activation functions is that they are located in the non-negative quadrant, that is, truncated and translated along the x-axis as compared to the standard sigmoid.

Typically, these LCLV activation functions are therefore represented by a set of non-negative x- and y-coordinates (response data) relating the write light intensity and the read-out light intensity of the LCLV. To be able to describe various properties (translation, range, gain) of the sigmoid-like LCLV response data and the differences with the standard sigmoid, a close approximation by a generic sigmoid curve fit is used, based on [5]. A generic sigmoid is an increasing, continuous, bounded, differentiable, S-shaped function with four real-valued parameters α, β ($gain$), γ, δ:

$$s(x) = \alpha + \frac{\gamma}{1 + e^{-\beta x + \delta}}. \tag{1}$$

Furthermore, for ease of integration in neural network simulations it is desirable that the upper bound of an activation function is scaled to unity, which can be obtained by scaling both the response data sets and their curve fits by dividing their y-values by their upper bound. The normalized curve fit parameters of the five LCLV activation functions are given in table 1. Response curves and curve fits for the first three LCLVs were obtained by Xue [5] [10]. The two LCLV4 curves are obtained from the same light valve [11], but by applying a voltage of approximately 10 volts for LCLV4a and 15 volts for LCLV4b [8]. The parameters for the standard sigmoid have been included in the first column of this table as a point of reference.

	standard sigmoid	LCLV1	LCLV2	LCLV3	LCLV4a	LCLV4b
α	0	−0.41	−2.34	0.015	−11.4	−0.019
β	1	0.087	0.0062	0.043	0.79	1.40
γ	1	1.41	3.34	1.0	12.4	1.0
δ	0	0.93	−0.82	3.20	−2.31	4.57

Table 1: Sigmoid curve fit parameters for the standard sigmoid and the LCLV activation functions.

The five LCLV response curves and their curve fits are shown in figures 2 to 6, whereas in figure 7 the generic sigmoid function (1) is depicted. In general the response curves are truncated on $x \geq 0$ and $y \geq 0$, and these truncated parts are closely approximated by their curve fits (see [8] for a more detailed description).

Since the LCLV activation functions are only defined on non-negative values, the neuron inputs of the activation function have to be non-negative. In our adapted learning rule, negative inputs are therefore considered to be equal to zero in the forward propagation step. Finally, to perform simulations with the response data, that are given as a set of x- and y-coordinates, a continuous

approximation by linear interpolation is used. The derivative of the response data curve, which is needed in the backward pass of the backpropagation learning algorithm, is defined to be the derivative of this linear interpolation.

4 Benchmarks and Simulation Parameters

As a test of the capability of the backpropagation algorithm [12] to train a three-layer network using activation functions realized by LCLVs, five different benchmarks have been used in the simulations. Three of them are well-known artificial problems, namely the exclusive or (XOR) problem, the 3-bit parity problem (Par), and the 4-bit addition problem (Add), where the modulo 2 sum of 2 numbers of 2 bits has to be calculated. Furthermore, two real-world data sets have been used, namely the sonar benchmark [13] and the wine data set [14]:

Sonar This data set was originally used by R. Gorman and T. Sejnowski in their study of the classification of sonar signals using a neural network. The task is to discriminate between sonar signals bounced off a metal cylinder and those bounced off a roughly cylindrical rock. Each pattern is a set of 60 numbers in the range $[0, 1]$. The corresponding output patterns are the two unit vectors.

Wine is the result of a chemical analysis of wines grown in a region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. A wine has to be classified using these 13 values, which have been scaled to the interval $[0, 1]$. The target patterns are the three unit vectors.

Table 2 shows the simulation parameters for the various benchmarks. Training is continued until the convergence criterion is satisfied or the maximal number of iterations has been performed. The convergence criterion used in the simulations is the minmax error: for all patterns in the training set the difference between the value of an output neuron and its target value has to be at most ε. In fact, this criterion has been slightly refined, to take into account that the minimal value of the LCLV curves differs slightly from zero . Therefore, all zero-valued targets in the benchmark sets have been replaced by these minimal values to obtain a fair comparison between the different curves.

Many experiments have been performed with these five benchmarks, varying the initial weights and the number of hidden units. Due to space limitations only the results for the XOR problem with 4 hidden units and the Sonar benchmark with 14 hidden units will be given in tabular form. These results are the mean over the converged experiments out of 20 (for XOR) and 5 (for Sonar) different random weight initializations in the interval $[-1, 1]$.

Benchmark	XOR	Parity	Addition	Sonar	Wine
max. # of epochs	10000	10000	10000	10000	10000
# of training patterns	4	8	16	104	178
# of inputs	2	3	4	60	13
# of outputs	1	1	2	2	3
# of hidden units	2–20	2–20	2–20	8&14	4&6&9
learning rate	0.3	0.3	0.1	0.1	0.6
ε	0.1	0.1	0.1	0.3	0.1

Table 2: Simulation parameters for the benchmarks.

5 Adaptation Rules and Simulation Results

A series of simulations has been performed with the backpropagation algorithm on a three-layer network, using both LCLV curve fits and LCLV response data as activation functions. The basic

	sigmoid		shifted sigmoid		LCLV1		LCLV2		LCLV3		LCLV4a		LCLV4b	
	mean	%	mean	%	mean	%	mean	%	mean	%	mean	%	mean	%
XOR	1998.3	100	1119.3	100	-	0	-	0	-	0	2842.1	35	588.6	100
Sonar	1199.6	100	888.2	100	-	0	-	0	-	0	-	0	512.2	100

Table 3: Results for the sigmoids and the LCLV curve fits: average number of iterations and percentage of converged experiments.

algorithm for all simulations was the on-line backpropagation learning rule, in which the weights are updated after each presentation of a pattern. As a point of reference, a series of simulations has been performed with the standard sigmoid and on-line backpropagation (see the first column of table 3). First, it will be shown how to handle activation functions which are located in the non-negative quadrant.

5.1 Adaptation of Weight Initialization

The initial weights for a multilayer neural network are usually uniformly chosen in an interval symmetric around zero [15]. However, this initialization method leads to non-convergence results when using a sigmoidal activation function which has been translated along the x-axis. An example of such functions are the LCLV activation functions described in section 3, but also in analog electronic implementations the sigmoid non-linearity can be translated along the x-axis [16]. The following weight initialization method resulting in neuron inputs centered around the inflection point of the activation function, has been used instead: (i) the weights and biases are chosen randomly with values in an interval symmetric around zero and (ii) to each weight from a neuron in layer ℓ to a neuron in layer $\ell+1$ and to each bias of a neuron in layer $\ell+1$ a term is added $shift/(N_\ell + 1)$, where $shift$ is the x-coordinate of the inflection point of the activation function, and N_ℓ the number of neurons in layer ℓ. In this way, neuron inputs are proportionally shifted towards the inflection point of the activation function. The inflection point of the curves is defined as the x-coordinate corresponding to a function value of a 0.5. The restriction of the activation function to the non-negative quadrant is taken into account by rounding negative neuron inputs to zero. This is the basic learning rule for all the experiments described in the rest of this paper.

5.2 Simulation Results

As a first test, a series of experiments has been performed with a standard sigmoid that has been translated four units along the positive x-axis and truncated to the positive quadrant which led to good results (see the second column of table 3) comparable to the ones obtained with the standard sigmoid. These results confirm that our adapted backpropagation learning rule can compensate for shifted and truncated sigmoidal activation functions.

Next, the adapted learning rule was used to train the networks using the five LCLV curve fits (see table 3). Only the results for the almost symmetric LCLV4b curve fit were good, and comparable to the results for the shifted sigmoid, while for the LCLV4a curve fit moderate results were obtained for the small artificial benchmarks. For the other curve fits the training process did not converge at all for all the benchmarks. Note that the curve fits with a gain β close to one, namely LCLV4a and LCLV4b, gave the best results. The reason why the results for LCLV4a are worse than for LCLV4b might be the asymmetry of this activation function (see figure 5). This is confirmed by the activation values of the output neurons having a tendency towards one in most of the simulations.

5.3 Theorem for Gain Compensation

As noted in section 5.2, the gain of the activation function might have an important influence on the performance of the backpropagation learning rule. Many rules of thumb for choosing the learning rate and the initial weights (or initial weight range) are namely based on a sigmoid with gain equal

	Network M	Network N
Activation function	$\varphi(\beta x)$	$\varphi(x)$
Gain	β	1
Learning rate	η	$\beta^2\eta$
Weights	$\mathbf{w}$	$\beta\mathbf{w}$

Table 4: The relationship between activation function, gain, weights, and learning rate.

to 1. Those rules of thumb are not applicable anymore when the gain of the activation function differs greatly from one; leading to slow convergence or no convergence at all for the backpropagation algorithm. As can be seen in table 1, the gain β of the LCLV curve fits (and hence the response data) has values different from one, especially for LCLV1, LCLV2, and LCLV3. Saxena and Fiesler [8] suggest to divide the initial weights and the learning rate by the gain to obtain better results with activation functions with a non-standard gain.

Using this heuristic the results for the activation functions with a small gain improved considerably, especially those for LCLV1 and LCLV3 that give good results for the three artificial benchmarks. Upon closer scrutiny, however, the gain parameter still seems to have a large influence, as is indicated by the non-convergence results for the LCLV2 curve which has the smallest gain.

This influence can be eliminated by applying a recently proven simple and precise relationship that enables compensating for the non-standard gain in backpropagation neural networks by changing the learning rate and the initial weights [17]:

Theorem 1 Two neural networks M and N of identical topology whose activation function φ, gain β, learning rate η, and initial weights $\mathbf{w}$ are related to each other as given in table 4, are equivalent under the on-line backpropagation algorithm; that is, when presented the same pattern set in the same order, their outputs are identical.

An increase of the gain with a factor β can therefore be compensated for by dividing the initial weights by β and the learning rate by β^2. Some extensions of the above theorem for variations of the standard backpropagation algorithm are also given in [17]. In the experiments described in the next section, one of these variants has been used: flat spot elimination [18]. This technique adds a constant to the derivative of the activation function in the backward pass. If this technique is applied, theorem 1 holds if the constant d added to the derivative of the activation function in network N is equal to the constant c added to the derivative in network M divided by the gain: $d = c/\beta$. The factors with which the initial weights, learning rate, and flat spot constant have to be multiplied to compensate for the non-standard gain β of the five LCLV curves are given in table 5.

Note that this technique is applicable to any activation function with a non-standard gain. From an engineering point of view this is pertinent since it opens up new device possibilities for non-linearities in neural networks. For example, the problem of training neural networks with high gain thresholds, which are efficient to implement in analog electronic hardware and use minimal power [19], has been eliminated.

	LCLV1	LCLV2	LCLV3	LCLV4a	LCLV4b
initial weights	11.49	161.3	23.26	1.27	0.71
learning rate	132.0	26.02 10^3	541.0	1.61	0.50
flat spot	0.087	0.0062	0.043	0.79	1.40

Table 5: Multiplication factors for gain adaptation.

5.4 Simulation Results

Extensive experiments were done for both the curve fits (table 6) and for the response data (table 7) while applying the gain theorem. The results for the curve fits for the three artificial benchmarks

	LCLV1		LCLV2		LCLV3		LCLV4a		LCLV4b	
	mean	%	mean	%	mean	%	mean	%	mean	%
XOR	973.2	100	2200.1	90	1222.2	100	3149.5	65	1265.2	100
Sonar	446.0	100	1019.8	100	864.8	100	-	0	1344.6	100

Table 6: Results with the gain theorem and the LCLV curve fits: average number of iterations and percentage of converged experiments.

	LCLV1		LCLV2		LCLV3		LCLV4a		LCLV4b	
	mean	%	mean	%	mean	%	mean	%	mean	%
XOR	561.4	95	636.4	100	896.5	100	669.9	95	933.2	100
Sonar	231.2	100	181.8	100	340.8	100	769.8	100	488.2	100

Table 7: Results with the gain theorem and the LCLV response data: average number of iterations and percentage of converged experiments.

are good and often comparable to the results with a shifted sigmoid, although the asymmetry of the LCLV4a curve fit still seems to be a problem and gives results that are slightly worse. This is confirmed by the results for the two real-world data sets which are good, except again those for the LCLV4a curve fit.

In the final experiments with the LCLV response data the flat spot elimination technique has been used to compensate for the flat region of the activation function, which is due to the linear interpolation on the measured data. In this way good results were obtained for all the LCLV curves and on all five benchmarks. This is illustrated by the results given in table 7, which even show a faster convergence of the training procedure than with the standard sigmoid.

6 Conclusions

An adapted backpropagation learning rule is presented here that compensates for the differences between non-standard activation functions as available in hardware and the standard sigmoidal activation function, since backpropagation usually performs poorly with these hardware activation functions. The adaptation consists in modifying the initial training conditions as well as compensating for the gain of the activation function. Especially the ability to handle arbitrary gains proves to be of great importance. Two methods to compensate for the gain have been applied to an optical multilayer neural network with non-linear thresholding by LCLVs. Simulations have been performed on five benchmark problems, both with the LCLV curve fits and their sampled response data. These experiments show that our adapted learning rule, based on a precise relation between the gain and the other initial parameters, performs well with five different optical activation functions realized by off-the-shelf liquid crystal light valves.

References

[1] F. Yu, T. Lu, X. Yang, and D. Gregory, "Optical Neural Network with Pocket-Sized Liquid-Crystal Televisions," Optics Letters, volume 15, number 15, pages 863–865, (1990).

[2] J.-S. Jang, S.-G. Shin, S.-W Yuk, S.-Y Shin, and S.-Y. Lee, "Dynamic Optical Interconnections Using Holographic Lenslet Arrays for Adaptive Neural Networks," Optical Engineering, volume 32, number 1, pages 80–87, SPIE, Bellingham, Washington, ISSN 0091–3286, (1993).

[3] I. Shariv and A. A. Friesem, "All-Optical Neural Network with Inhibitory Neurons," Optics Letters, volume 13, pages 485–487, (1989).

[4] D. Psaltis and Y. Qiao, "Adaptive Multilayer Optical Networks," In *Progress in Optics*, E. Wolf (editor), volume 31, chapter 4, pages 227–261, Elsevier Science Publishers, Amsterdam, The Netherlands, ISBN 0–444–89836–0, (1993).
See also: D. Psaltis and Y. Qiao, "Optical Neural Networks," Optics & Photonics News, pages 17–21, (1990).

[5] W. Xue, "Characterization of Liquid Crystal Light Valves for Neural Network Applications," PhD thesis, IMT, University of Neuchâtel, Neuchâtel, Switzerland, (1994).

[6] M. G. Robinson and K. M. Johnson, "Noise Analysis of Polarization-Based Optoelectronic Connectionist Machines," Applied Optics, volume 31, number 2, pages 263–272, (1992).

[7] W.P. Bleha, L.T. Lipton, E. Wiener-Avnear, J. Grinberg, P.G. Reif, David Casasent, H.B. Brown, and B.V. Markevitch, "Application of the Liquid Crystal Light Valve to Real-Time Optical Data Processing," Optical Engineering, volume 17, number 4, pages 371–384, (1978).

[8] I. Saxena and E. Fiesler, "Adaptive Multilayer Optical Neural Network with Optical Thresholding," Optical Engineering (ISSN 0091-3286), special on Optics in Switzerland (P. Rastogi, editor), volume 34, number 8, pages 2435–2440, (1995).

[9] E. Fiesler, " Neural Network Classification and Formalization," Computer Standards & Interfaces, special issue on Neural Network Standards, John Fulcher (editor), volume 16, number 3, pages 231–239, North-Holland / Elsevier Science Publishers, Amsterdam, The Netherlands, ISSN 0920-5489, (1994).

[10] N. Collings and W. Xue, "Liquid Crystal Light Valves as Thresholding Elements in Neural Networks: Basic Device Requirements," Applied Optics, volume 33, number 14, pages 2829–2833, (1994).

[11] Micro-Optics Technologies, 8608 University Green # 5, Middleton, WI 53562, U.S.A.

[12] D. Rumelhart, G. Hinton, and R. Williams, "Learning Internal Representations by Error Propagation," Parallel Distributed Processing: Explorations in the Microstructure of Cognition, volume 1: Foundations, pages 318–362, MIT Press, Cambridge, Massachusetts, ISBN: 0–262–18210–7, (1986).

[13] R. P. Gorman and T. J. Sejnowski, "Analysis of hidden units in a layered network trained to classify sonar targets," Neural Networks, volume 1, pages 75–89, (1988).

[14] P. M. Murphy and D. W. Aha (Librarians), "UCI Repository of Machine Learning Databases," Machine-readable data repository accessible via anonymous ftp ics.uci.edu: pub/machine-learning-databases, (1994).

[15] G. Thimm and E. Fiesler, "Weight Initialization in Higher Order and Multi-Layer Perceptrons," IEEE Transactions on Neural Networks, conditionally accepted for publication.
See also: G. Thimm and E. Fiesler, "Weight Initialization in Higher Order and Multi-Layer Perceptrons," Proceedings of the '94 SIPAR–Workshop on Parallel and Distributed Computing, M. Aguilar (editor), pages 87–90, Institute of Informatics, University of Fribourg, Fribourg, Switzerland, (1994).

[16] J. M. C. Oosse, H. C. A. M. Withagen, and J. A. Hegt, "Analog VLSI Implementation of a Feed-Forward Neural Network," Proceedings of the First International Conference on Electronics, Circuits, and Systems (ICECS'94), Cairo, Egypt, (1994).

[17] G. Thimm, P. Moerland, and E. Fiesler, "The Interchangeability of Learning Rate and Gain in Backpropagation Neural Networks," to appear in volume 8, number 2 of *Neural Computation*, (1996).

[18] S. E. Fahlman, "An Empirical Study of Learning Speed in Backpropagation Networks," Technical Report CMU-CS-88-162, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, (1988).

[19] D. B. Mundie and L. W. Massengill, "Threshold Non-Linearity Effects on Weight-Decay Tolerance in Analog Neural Networks," Proceedings of the International Joint Conference on Neural Networks (IJCNN'92), volume 2, pages 583–587, Baltimore, USA, (1992).