

Enhancing Performance of Relational Fuzzy Neural Networks with Square BK- Products

Warren L. Davis, IV and Ladislav J. Kohout

Department of Computer Science, Florida State University, Tallahassee, Florida 32306-4530, USA.

E-mail: wdavis@cs.fsu.edu, kohout@cs.fsu.edu

Abstract—In this paper, we extend research done in max-min fuzzy neural networks in several important ways. We replace max and min operations used in the fuzzy operations by more general t-norms and co-norms, respectively. In addition, instead of the Łukasiewicz equivalence connective used in network of Reyes-Garcia and Bandler, we employ in our hybridization a variety of equivalence connectives. We explore the effectiveness of this network in the domain of phoneme recognition, and diabetes data. We find increased classification ability in many cases, as well as great potential for further expansion of the use of fuzzy operations in the field of pattern recognition.

I. INTRODUCTION

Neuro-fuzzy hybridization methods can be grouped broadly in two categories [1]: (i) as fuzzy neural networks (FNN), where a neural network is equipped with the capability of handling fuzzy information, and (ii) as neuro-fuzzy systems (NFS) involving a fuzzy system augmented by neural networks to enhance some of its characteristics. We focus on the first method of hybridization, where neurons are designed to perform various operations used in fuzzy set theory (like fuzzy union, intersection, aggregation etc.) instead of the standard multiplication addition operations.

In this paper we further extend the method of papers [2] and [3]. In these papers, Reyes-Garcia and Bandler augment the fuzzy max-min network developed by Pedrycz [4]-[6] by incorporating the fuzzy BK-products of relations [7]-[9] into the FNN system.

We introduce two extensions: (i) we replace max and min operations by more general t-norms and co-norms respectively; (ii) instead of the BK-square product of relations based on the Łukasiewicz equivalence connective as used by Reyes-Garcia and Bandler, we employ in our hybridization a variety of equivalence connectives, some of which, unlike the Łukasiewicz one, are not bi-residuals.

II. FUZZY MAX-MIN NEURAL NETWORK

A. Pedrycz original Neural Network

A classical feed-forward single layer perceptron consists of a collection of input nodes $X = \{x_1, x_2, \dots, x_m\}$, a collection of output nodes $Y = \{y_1, y_2, \dots, y_n\}$, and a weight matrix $W = \{w_{ij} | i \in m, j \in n\}$. The output at node $y_j = f(\sum_i x_i w_{ij})$ is a function of its summed, weighted inputs. A bias node x_0 can also be included.

Pedrycz [4]-[6] replaces these structures by fuzzy relations. The weight matrix of the classical perceptron is replaced by a

fuzzy relational matrix that represents a two- argument fuzzy relation from X to Y , $R = \{xRy | x \in X, y \in Y\}$, so that the connection between x_i and y_j has a relational value $R(x_i, y_j)$.

In this fuzzy neural network, the output values for Y are generated by the compositional fuzzy product of the set of inputs X and the relations R , or $Y = X \circ R$. This, written explicitly, is $Y(y_j) = \max x_i(\min(X(x_i), R(x_i, y_j)))$.

A fuzzy equality index based on the Łukasiewicz implication provides a performance metric for evaluation of these fuzzy-relational classifications in training. This metrics can be used in a gradient-descent manner to update the relational matrix in a similar fashion to the standard back-propagation algorithm. For derivation of the gradient descent algorithm see [4]. Most importantly, Pedrycz also shows, that the XOR problem, a problem unsolvable by the standard single-layer network [10] is solvable and converges with the relational single-layer network proposed in [4].

B. The Reyes-Garcia and Bandler Modifications

1) *An Overview*: The system architecture proposed by Carlos Reyes-Garcia and Wyllis Bandler using a fuzzy relational neural network as a pattern classifier was adopted from the general classifier presented by Pedrycz in [4].

The significant departure from the previous work of Pedrycz was the exploration of a BK-relational square product for the classification of unknown patterns. The system was tested on a speech recognition problem where membership values were calculated with the *pi* and *dense-trapezoidal* functions [2],[3]. The use of Nn -dimensional vectors to represent the fuzzy membership values that was needed for a linguistic application was inspired by the work presented by Pal in [11], [12]. Carlos Reyes-Garcia and Wyllis Bandler used this network model, with modifications, to perform phoneme recognition on the Kohonen phoneme data set [2],[3].

2) *Learning Phase*: The learning phase of the speech recognition system developed by Reyes-Garcia and Bandler uses supervised learning to train the network from processed data samples. The Linguistic Feature Extractor (LFE) that fuzzifies the data samples into N classes per feature performs this initial processing. These N classes would correspond, for instance, to grades of membership such as low, medium, and high for $N=3$. For n input features, the linguistic feature extractor would create an input vector of cardinality Nn .

After the classification of the input features, the Desired Output Estimator (DOE) calculates the desired class member-

ships to each of the output classes given the input vector. The class memberships are made less ambiguous by fuzzy ENH operation. The Neural Network Trainer then uses these desired membership values to train the network using the learning algorithm introduced in [4].

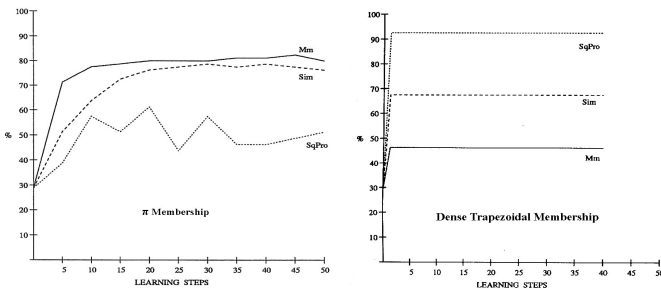
3) *Processing Phase*: The processing phase begins by using the same LFE defined in the learning phase. The LFE in the processing phase uses the parameters established by the LFE in the learning phase, and operates in a similar way to fuzzify the testing (processing) data, creating a Nn -dimension Linguistic Properties Vector (LPV).

The Fuzzy Classifier (FC) then uses the relational matrix developed during the learning phase and the LPV to process the input patterns. The patterns were processed using three different compositional products: max-min \circ , GeometricMean (Sim) $Y(y_j) = \max_{x_i}(\min(X(x_i), \sqrt{R(x_i, y_j)}))$, and square BK-product $(Y \square R)(y_j) = \max_{x_i}[X(x_i) \equiv R(x_i, y_j)]$.

These three relational products were tested in conjunction with the π and dense trapezoidal membership functions. The equivalence connective used by Reyes-Garcia [2],[3] was based on the Łukasiewicz implication.

The network was trained and tested using the vowels {a,e,i,o,u} in the Kohonen data sets. The correct classification percentages (CCP) are shown in graphs in Figure 1 below, reprinted (with modified labels) from [2],[3].

There is a significant difference in performance between the π and trapezoidal membership functions. When the π membership function was used, the best results were achieved with original max-min compositional rule. As can be seen in Figure 1, it reached its highest performance when the network was allowed to train for 45 learning steps with a CCP of about 86.41% on the testing set. The other compositional products did not train as quickly, nor reach as high of a CCP during the testing.



The results with the dense trapezoidal membership function, however, were quite different. As can be seen in Figure 1, the BK-relational square product was the highest performer, with a CCP of 93.14%. The modified compositional product came in second, and the standard max-min composition gave the lowest performance. In addition, all three compositional products reached their maximum CCP when the network was trained for 2 epochs, considerably less than with the membership function. This is very important for larger scale recognition problems. In addition, the training rates were completely stable

after those 2 epochs; additional training produced no effect on performance for the testing sets.

These are very interesting results that led us to continue where Reyes-Garcia and Bandler left off and embark on a study of these networks more systematically.

III. OUR EXPERIMENTATION WITH THE BK-SQUARE PRODUCT IMPLEMENTATION

A. Extension of Garcia-Reyes and Bandler's results

It is the surprising and somewhat inexplicable synergy between the trapezoidal membership function and the BK-relational square product [7]-[9] that is the focus of this research. There are many questions concerning the nature of this combination that are being explored through various methodologies. To explore this more in-depth, the original software used in [2],[3] was modified to allow for expanded experimentation on the interaction between the fuzzy logical operators. Other than the logical operator combinations, all other parameters were kept the same as in the Reyes-Garcia and Bandler research.

1) *Varying the BK-Square Product Implication Operators*: In [2], [3], the BK-relational square product was used in the processing phase of the neural network. The output at a node y_j in Y , was dependent on the i inputs in X and the relationships $R(x_i, y_j)$ such that

$$Y(y_j) = (X \square R)(y_j) = \min_{x_i}(X(x_i) \leftrightarrow R(x_i, y_j)) \quad (1)$$

Equivalence is formed by a conjunction of implication and co-implication, $a \leftrightarrow b = (a \rightarrow b) \wedge (a \leftarrow b)$

The implication used in [2], [3] for these equations was the Łukasiewicz implication and conjunction \wedge was Minimum. However, there are many implication operators in widespread use today, some better describing different domains than others. For this research, a total of 22 different implication operators were tested, all varying in complexity and fuzziness. The manner in which these different implications handle the antecedent and consequent, the congruity of the implication with the crisp logic counterpart, and other implication properties can vary widely, as discussed in great detail in [8]. Because of this, their classification of the testing data often varies substantially.

2) *Varying the T-norms and T-conorms within implications*: Many of the implications implemented to form the square product are defined in terms of maximum and minimum values. The Kleene-Dienes implication, for instance, is defined as $x \rightarrow y = \max(1 - x, y)$. In these cases, the minimum and maximum can often be seen as specific instances of t-norms and t-conorms, varying representations of the intersection and union operators, respectively. Although the original representation for these implications was in terms of the min and max t-norms and t-conorms, it seems plausible that other t-norms and t-conorms could be substituted for these effectively, creating new implications and thereby new equivalences. To this end, the implications were implemented as generic t-norms and t-conorms, and several prominent t-norms and t-conorms were added to the software to be used in testing.

It is noted that, for some cases, certain substitutions could yield invalid results, such substitution of the aforementioned t-norms in the Gaines-43 and Łukasiewicz implications. These combinations are those in which the t-norm or t-conorm is substituted for the operators min and max enforcing the *boundary conditions* ensure that the value of the implication stays between 0 and 1. Nevertheless, for completeness, these combinations are also implemented as an option in the software. This will allow us to perform experiments with tall fuzzy sets [21] that have values outside [0, 1].

3) *Varying the conjunction operator between implications:* Equation (5) shows that the equivalence is defined as the minimum of the implication and the co-implication. Once again, this minimum can be seen as a specific instance of the t-norm operator. Therefore this was also implemented as a generic t-norm in the software. This substitution is independent of that which may be used within the implications.

$$\begin{aligned} EqConj^{\min} &= \min((x \rightarrow y), (y \rightarrow x)) \\ EqConj^{prod} &= (x \rightarrow y) \cdot (y \rightarrow x) \\ EqConj^{Luka} &= \max(0, (x \rightarrow y) + (y \rightarrow x) - 1) \\ EqConj^{drastic} &= \begin{cases} (x \rightarrow y) & \text{if } (y \rightarrow x) = 1 \\ (y \rightarrow x) & \text{if } (x \rightarrow y) = 1 \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

If continuous t-norms and s-norms are used, the substitutions have always values in [0, 1], because Min is the largest such t-norm, and Max is the smallest such s-norm (i.e. t-conorm).

4) *Varying the aggregation of equivalences:* Equation (6) shows that the square product is defined as the minimum of the equivalence relationships that are being aggregated, listed in this research as $SqProd^{\min}$.

Another valid representation of the square product is the mean over the set of equivalences. This gives the following node output:

$$Y(y_j) = \frac{1}{n} \sum X(x_i) \leftrightarrow R(x_i, y_j) \quad (2)$$

which is termed the *mean* square product aggregation, and listed as $SqProd^{mean}$. Outliers can be *trimmed* from the set before averaging using a method as defined in [13]. This aggregation is listed as $SqProd^{trim}$.

IV. RESULTS AND ANALYSIS

A. Kohonen Phoneme Data

In total, 1248 different implementations of the BK-relational square product were tested for two datasets. The first is phonetic data from research done by Kohonen et al. at the University of Helsinki and available as part of the SOM/LVQ_PAK [14], a widely studied package of phoneme data and processing utilities. This was the data used by Reyes-Garcia and Bandler in their original experimentation. Some of the highest results are listed below:

\rightarrow	T-norm	S-norm	$\leftrightarrow (\wedge)$	Correct %
EZ	Łuk.	Probabl.	Łuk.	94.6164
EZ	Drastic	Probabl.	Łuk.	94.4818
EZ	Min	Max	Product	94.3472
EZ	Product	Max	Product	94.3472
Bot (m_5)	Min	Max	Min	94.3472

$\leftrightarrow (\wedge)$ denotes \wedge used to make \leftrightarrow from implications. The

overall highest performing implications were formula (3) the Early Zadeh and formula (4) – Bot (m_5) i.e. Lower Bound counterpart of Wilmott (4), see ref. [15]. It should be noted that each of these formulae in their original forms were written using the minimum t-norm and the maximum s-norm. They are written below in their generic implementations as introduced in this research:

$$x \rightarrow y = S(T(x, y), (1 - x)) \quad (3)$$

$$x \rightarrow y = T\left(\begin{matrix} S((1 - x), x + y - 1), \\ S(y, 1 - x - y) \end{matrix}\right) \quad (4)$$

where T and S represent generic t-norms and s-norms.

As a comparison the original network developed by Reyes-Garcia and Bandler achieved at an accuracy of 93.14% [2], [3]. Of the 1248 test combinations of various fuzzy connectives implemented in our research, 412 achieved better results, 186 achieved the same percentage, and 650 achieved lesser results.

B. Pima Indians Diabetes Data

Another data set yielding interesting results was that of the Pima Indians Diabetes data from the Proben1 Benchmark suite [16]. The Pima Indians diabetes dataset consists of data used to identify cases of diabetes from a 768 samples by using 8 diagnostic features. It is part of the Proben1 data suite [16], and was used in this research following the guidelines of the Proben1 benchmarking methodology. The network fuzzified the data into 7 linguistic features, trained over 576 samples with 2 epochs of training, and was tested on 192 samples.

The Gödel implication performing the highest, yielding 81.7708% accuracy. Incidentally, this implication did not perform well at all with the Kohonen phoneme data.

These results compare favourably to results of other researchers on the same dataset shown in the table below.

Research	% Accuracy
Rutkowski & Cpalka	77.8
Au & Chan	77.6
Ster & Dobnikar	77.7
Smith et al.	76.0
Our Results	81.8

See Rutkowski and Cpalka [18], Au & Chan [20], Smith et al. [17] using ADAP algorithm, Ster and Dobnikar [19] using various methods, Logdisc 77.7% including Kohonen self-organizing maps (72.2%) and LVQ (75.8%).

One of the more interesting aspects of these findings is how a change in the selection of t-norm, s-norm, or other parameter can greatly affect the performance of the equivalence relations in the network. For instance, the Early Zadeh implication performs decently well with the T^{\min} t-norm, the S^{\max} s-norm, the $EqConj^{\min}$ equivalence conjunction, and the $SqProd^{\min}$ aggregation, with a CCP of 93.14%. However, with these same settings and using $EqConj^{drastic}$ the network performs much worse, with a CCP of 29.34%, the result of lumping all phonemes into the first class (phoneme ‘‘A’’).

One parameter that consistently produces a large change in classification ability is the square product

aggregation. For instance with the implementation $\{Klir - Yuan, EqConj^{drastic}, SqProd^{min}\}$, a very low CCP, 29.34%, was obtained. However, by switching the square product aggregation to $SqProd^{trim}$, the CCP jumped to 94.0781%.

V. CONCLUSIONS AND CURRENT AND FUTURE RESEARCH

It is clear that deviation from the standard implementation of implications, equivalence, and consequently the compositional relational products that use them, can produce different results given the fuzzy membership values entered into them. In the fuzzy max-min relational model, these varying choices of logic connectives for the BK-relational square product produced a wide range of accuracies. In the phonetic recognition example, accuracy ranged from 94.6164% all the way down to 8.61373% for different connectives.

Research into the performance of the BK-relational square product within the Pedrycz type of relational network continues. The natures of the highest and lowest performers are being analyzed mathematically in relation to the functionality of the relational network, the membership function, and the properties of the data being used. It has also been noted that square product implementations train at different rates. Although the implementation in the Reyes-Garcia and Bandler research trained in 2 epochs, other implementations may benefit from fewer or more epochs of training. In addition, the reduction of training epochs (from 45 epochs down to 2) [2], [3], provided by the method described in this paper may be very important in some applications (such as recognition of satellite pictures) that are hindered by rather longer training required for more conventional neural networks.

VI. REFERENCES

- 1) Sankar K. Pal and Sushmita Mitra. *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing*. John Wiley & Sons, 1999.
- 2) Carlos Alberto Reyes-Garcia, *On the Design of a Fuzzy Relational Neural Network for Automatic Speech Recognition*. Ph.D. Dissertation, Dept. of Computer Science, Florida State University, Tallahassee, FL., Spring 1994.
- 3) Carlos Alberto Reyes-Garcia and Wyllis Bandler, "Implementing a Fuzzy Relational Network for Phonetic Automatic Speech Recognition," *Fuzzy Modeling: Paradigms and Practice*. Ed. Witold Pedrycz. The Kluwer International Series in Intelligent Technologies, Vol. 7. Kluwer Academic Publishers. 1996.
- 4) Witold Pedrycz, "Neurocomputations in Relational Systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, No. 3, March 1991, pp 289-296.
- 5) Pedrycz, W. "Fuzzy Neural networks with reference neurons as pattern classifiers," *IEEE Transactions on Neural Networks*, vol. 3, 1992.
- 6) Pedrycz, W. and Rocha A. F. "Fuzzy-set based models of neurons and knowledge-based networks." *IEEE Transactions on Fuzzy Systems*. FS-1. 1993.
- 7) W. Bandler and L. J. Kohout, *Mathematical Relations, their Products and Generalized Morphisms*. Tech. report, Man-Machine Systems Laboratory, WWS-MMS-REL 77-3, Dept. of Electrical Engineering, University of Essex, Colchester, Essex, U.K., 1977.
- 8) Wyllis Bandler and Ladislav Kohout, "Fuzzy Power Sets and Fuzzy Implication Operators," *Fuzzy Sets and Systems*, vol. 4, North Holland Publishing Company, 1980, pp. 13-30.
- 9) Ladislav J. Kohout, "Boolean and Fuzzy Relations", *Encyclopedia of Optimization, vol. I (A-D)*, C.A. Floudas and P.M. Pardalos Ed., Kluwer Academic Publishers, 2001, pp. 189-202.
- 10) M. Minsky and S. Papert, *Perceptrons*. MIT Press, 1969.
- 11) S. K. Pal and Mandal D. P., "Linguistic Recognition Systems Based on Approximated Reasoning," *Information Science*, vol. 61, No. 2, 1992, pp. 135-161.
- 12) S. K. Pal, "Multilayer Perceptron, Fuzzy Sets, and Classification," *IEEE Transactions on Neural Networks*, vol. 3, No 5, Sep 1992, pp 683-697.
- 13) Jay L. Devore, *Probability and Statistics for Engineering and the Sciences*, California Polytechnic State University, 1995.
- 14) Teuvo Kohonen, et. al., "LVQ_PAK: The Learning Vector Quantization Program Package." Version 3.1. Report A30, Helsinki University of Technology, Faculty of Information Technology, Laboratory of Computer and Information Science, 1996.
- 15) Wyllis Bandler and Ladislav J. Kohout. "Semantics of Implication Operators," *International Journal of Man-Machine Studies*, vol. 12, 1980, pp 89-116. Reprinted in E. H. Mamdani and B. R. Gaines (eds.), *Fuzzy Reasoning and its Applications*, London, etc.: Academic Press, 1981, pp 219-246.
- 16) Lutz Prechelt, "Proben1- A Set of Neural Network Benchmark Problems and Benchmarking Rules." Technical Report, Fakultat fur Informatik, Universitat Karlsruhe, number 21/94, 1994.
- 17) Smith, J.W. (1988). "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus." In *Proceedings of the Symposium on Computer Applications and Medical Care* (pp. 261-265). IEEE Computer Society Press.
- 18) Rutkowski, L., and Cpalka, K "Flexible weighted neuro-fuzzy systems" *IEEE Transactions on Neural Networks*, Vol. 14, No. 3, May 2003.
- 19) B. Ster and A. Dobnikar, "Neural networks in medical diagnosis: Comparison with other methods," in *Proc. Int. Conf. EANN'96*, A. Bulsari, Ed., 1996, pp. 427-430.
- 20) W. H. Au and K.C.C. Chan, "Classification with Degree of Membership: A Fuzzy Approach," in *Proc. of the 1st IEEE Int'l Conf. on Data Mining*, San Jose, CA, 2001.
- 21) Mareš, M., *Computation over fuzzy quantities*, Boca Raton : CRC Press, 1994.