GENERALIZED NEURAL NETWORKS FOR FUZZY MODELING

K.-H. Temme, R. Heider and C. Moraga

Department of Computer Science, Computer Engineering and Computing Education
University of Dortmund
44221 Germany

{temme | heider | moraga}@LS1.informatik.uni-dortmund.de

Summary

Neuro-fuzzy modeling has been intensively studied since the early nineties. Recently a method has been disclosed, that uses a classical feedforward neural network with just one hidden layer. Nodes of the hidden layer use the logistic function as activation function meanwhile the output node has a linear activation function. This paper introduces a generalization of the logistic function and evaluates its capabilities with respect to neuro-fuzzy modeling. It is shown that a product-generated symmetric summation provides an exact interpretation of the activity of each node of the hidden layer.
Keywords: Neuro-fuzzy-modeling, aggregation

1. INTRODUCTION

Fuzzy modeling based on neural networks is a well established area within Computational Intelligence. Most work has been done using RBF nets [16] or special feedforward networks, where the first hidden layers realize the fuzzification (association of physical crisp inputs to linguistic terms) and compute the corresponding degree of satisfaction. Further layers compute the fuzzy inference and, if needed, additional layers will cope with defuzzification. Probably one of the best known representatives of this class is the ANFIS system [13], since it is presently included in Matlab, however similar networks have been discussed in the literature (see *e.g.* [6], [8], [11], [18], [19], [20]).

A recent contribution [1] has opened very attractive possibilites for effectively applying the simplest classical feedforward neural network for fuzzy modeling. The neural network has only one hidden layer where the processing nodes have a logistic activation function $f(wsi) = (1+\exp(-wsi))^{-1}$, ("wsi" denotes here "weighted sum of inputs"), and a linear output node. A neural network of this kind satisfies the conditions for universal approximation [9], [15] which is roughly equivalent to saying that for any physical system there exists at least one neural network of this kind, which can identify the system with arbitrary precision if enough representative performance examples are available. Furthermore it has been shown [1] that these neural networks, under an especial interpretation (see below) are equivalent to additive fuzzy systems [13].

The present paper introduces a generalization of these neural networks however preserving their interpretability as additive fuzzy systems.

The rest of the paper is organized as follows. In the next section an extended class of activation functions will be discussed and their characteristic properties will be explained. Furthermore it will be shown, that the method introduced in [1] may be applied to the new neural networks making them appropriate for fuzzy modeling. An example followed by a summary and conclusions will close the paper.

2. GENERALIZED LOGISTIC FUNCTIONS

Definition 1:

A generalized logistic function $f: R \to [0,1]$ is given by:

$$f(x) = \frac{1}{1 + k^{-x}} \qquad (1)$$

where $k \in R$ and $k > 1$.
It becomes apparent that the classical logistic function is a particular case of (1), when $k = e$. Equation (1)

represents a family of "soft step" functions whose steepness is controlled by the parameter k.

Observations:

- f(x) is overall differentiable with respect to x, *i.e.* the generalized logistic function may be used in neural networks trained with the backpropagation algorithm
- Equation (1) satisfies the conditions of continuity, $lim_{x\to-\infty}f(x) = 0$ and $lim_{x\to+\infty}f(x) = 1$, *i.e.* f(x) is a *squashing* function [9]. It follows that neural networks with one hidden layer using the generalized logistic function as activation function and an output linear node are universal approximators [9].
- Equation (1) is invertible. It follows that neural networks using *allover* the generalized logistic function as activation functions are universal approximators [15].
- The partial derivative of f(x) with respect to k is continuous and well defined, *i.e.* it is possible to extend the backpropagation algorithm to also train *the best k* at every node of the hidden layer. Similar work done to optimize the steepness of the classical logistic function showed that a faster learning and a better generalization could be achieved [7].

3. A FUZZY LOGIC INTERPRETATION

In [1] it was proven that the activity of a node of a neural network using a sigmoide activation function (k = *e* in the notation of the present paper) may be interpreted as a fuzzy *if-then* rule, if a special aggregation of the premises is used. In what follows it will be shown, that the same interpretation may be extended to the case of the generalized logistic function.

<u>Lemma 1</u>:

Let f be a generalized logistic function. There exists a bijection Ψ: (R, +) → ((0,1), *) producing the following transformation:

$$f(x+y) = f(x) * f(y) \qquad (2)$$

where the operation "*" will be expressed explicitly below (see eq. (3)).

Proof:

$$f(x+y) = \frac{1}{1+k^{-x-y}} = \frac{1}{1+k^{-x}k^{-y}}$$

but since

$$k^{-x} = \frac{1-f(x)}{f(x)} \quad \text{and} \quad k^{-y} = \frac{1-f(y)}{f(y)} \quad \text{then}$$

$$f(x+y) = \frac{1}{1+\left(\frac{1-f(x)}{f(x)}\right)\left(\frac{1-f(y)}{f(y)}\right)}$$

$$f(x+y) = \frac{f(x)f(y)}{(1-f(x))(1-f(y))+f(x)f(y)}$$

Define

$$f(x)*f(y) = \frac{f(x)f(y)}{(1-f(x))(1-f(y))+f(x)f(y)} \qquad (3)$$

The assertion of the Lemma follows.

The * operation is a *symmetric summation* [17]. Symmetric summations have the following general structure:

$$\forall\, x,y \in (0,1): \quad s(x,y) = \frac{g(x,y)}{g(1-x,1-y)+g(x,y)} \qquad (4)$$

where g is the *generating function*. It becomes apparent that in the case of the * operation, g is the product. Since for arguments in the open interval (0,1) the product is positive, continuous, commutative, non decreasing in both arguments and approaches 0 as the arguments approach 0, it follows [17] that the * operation is continuous, commutative and associative in the open interval (0,1).

From equation (1) it is simple to show that
f(-x) = 1 - f(x) and that f(0) = (1/2). This leads to:

f(x-x) = f(0) = ½
f(x-x) = f(x) * f(-x) = f(x) * (1-f(x))

i.e. ((0,1), *) is an Abelian group with 1/2 as neutral element and the inverse of an element f(x) is given bei 1- f(x) ∀ x ∈ R.

It should be noticed, that the operation * does not depend on the choice of *k*, but the choice of *k* affects the value of f(x) and f(y) in the former analysis.

It is interesting to mention that the * operation has also been considered by other authors. In [2] it is considered both as an associative symmetric summation and as a symmetric summation generated by the product. An interpretation as a uninorm is given in [4] meanwhile in [12] it is shown that * behaves as a Hamacher-2 t-norm in $[0, ½]^2$ and as a Hamacher-2 dual t-conorm in

[½, 1]^2. Finally in [14] * is considered as a non-linear convex combination of the t-norm product and its dual t-conorm.

The above proven Lemma 1 supports the following interpretation. Consider a node in the hidden layer of a neural network with n inputs. The operation of the node may be expressed as:

$$y = f\left(w_1 x_1 + w_2 x_2 + ... + w_n x_n\right)$$

where w_1 , ..., w_n are real-valued input weights to the node. After Lemma 1 the operation at the node may be given by

$$y = f(w_1x_1) * f(w_2x_2) * ... * f(w_nx_n) \qquad (5)$$

Let

$$f_{w_j}(x_j) =_{def} f(w_j x_j) = \frac{1}{1+k^{-w_j x_j}} = \frac{1}{1+(k^{w_j})^{-x_j}}$$

i.e. $f_{w_j}(x_j)$ is a generalized logistic function with parameter k^{w_j}. Then

$$y = f_{w_1}(x_1) * f_{w_2}(x_2) * ... * f_{w_n}(x_n). \qquad (6)$$

From the point of view of fuzzy logic, f_{w_j} may be seen as the characteristic function (fuzzy set) of a fuzzy linguistic term and $f_{w_j}(x_j)$ as the degree of membership of x_j to the fuzzy set f_{w_j}. Since the generalized logistic function represents a "soft step" fuzzy set, the corresponding linguistic term associated to that fuzzy set will be "at least <name of the fuzzy set>" if $w_j > 0$ otherwise "at most <...>". In analogy to the measurements of rise time of the step-answer of amplifiers, the value of x_j leading to $f_{w_j}(x_j) = 0.9$ will be taken as reference.

Accordingly, the fuzzy rule given by a hidden node of the neural network is the following:

if** x_1 is at least T_1 **and** x_2 is at least T_2 **and**... **and** x_n is at least T_n **then

$$y = f_{w_1}(x_1) * f_{w_2}(x_2) * ... * f_{w_n}(x_n)$$

where f_{w_j} represents the linguistic term *at least T_j* (assuming that $w_j > 0$).

It becomes apparent, that the output node of the neural network computes then the weighted sum of the former fuzzy rules thus modeling an additive fuzzy system [13].

The effect of a possible bias in a hidden processor upon the interpretation as a fuzzy rule should however still be analyzed. The generalized logistic function turns into:

$$f^{(b)}(x) =_{def} \frac{1}{1+(k)^{-(x-b)}} = f(x - b)$$

From Lemma 1 and from the associativity of * follows that:

$$f(x + y - b) = f(x)*f(y)*f(-b)$$

$$= \frac{f(x)f(y)f(-b)}{(1-f(x))(1-f(y))(1-f(-b)) + f(x)f(y)f(-b)}$$

but since "b" is a constant, $f(-b) = \frac{1}{1+k^b}$ and $1 - f(-b) = f(b) = \frac{1}{1+k^{-b}} = \frac{k^b}{1+k^b}$. Then:

$$f(x+y-b) = \frac{f(x)f(y)}{k^b(1-f(x))(1-f(y)) + f(x)f(y)} \qquad (7)$$

This equation may well be convenient for computation purposes, but it provides no simple interpretation. The following Lemma however solves this question.

<u>Lemma 2:</u>

$\forall v \in (0,1)$ let $\qquad \lambda^{(b)}(v) =_{def} \frac{v}{v - k^b(v-1)}$

Then holds:

$$f(x) * f(y) * f(-b) = \lambda^{(b)}\left(f(x) * f(y)\right) \qquad (8)$$

It is important to notice that $\lambda^{(b)}$ may be interpreted as a linguistic modifier, acting upon the * operation on the *non-biased* functions f(x) and f(y).

Example:

As an example, the well known test problem "Iris" [3] will be used. This example was processed in [1] to illustrate the translation of the neural network into a set of rules. This will not be repeated here. Only a comparison between a *classical* neural network with three hidden elementary processors with the logistic function as activation function including bias option, and a neural network with the same architecture but using the generalized logistic function excluding the bias option will be done. In both networks the output node uses a linear activation function. The “Iris” problem is known to have one linear separable class and two overlapping classes, thus being a non-easy problem for feedforward neural networks. By using an extended gradient descent algorithm both weights and k-values were adjusted to minimize the classification error. The obtained values were k_1 = 1.50544, k_2 = 2.44804 and k_3 = 1.53952 and by careful choice of the classification thresholds, (0.35 and 0.78, respectively) *a single* classification error was achieved considering both the training and the test set.

4. CONCLUSIONS

A generalization of the logistic function was introduced and its properties were analyzed. The function computed by each node of the hidden layer may be interpreted as a fuzzy if-then rule such that the resulting premises are aggregated by a symmetric sum. The slope of the generalized logistic function may be adjusted with a gradient descent procedure and this results in modifying the edges of the corresponding fuzzy linguistic terms.

Acknowledgements

The authors are very thankful to Enric Trillas and Ana Pradera for pointing out that the * operator is a symmetric summation and for providing relevant references on the subject. The authors appreciate the suggestions given by one of the Referees to improve the notation used in the paper.

References

[1] Benítez J.M., Castro J.L., Requena I. (1997). Are artificial neural networks black boxes? *IEEE Transactions on Neural Networks* **8**, 1156-1163.

[2] Dubois D., Prade H.(1985). A review of fuzzy set aggregation connectives. *Information Sciences* **36**, 85-121.

[3] Fisher R.A. (1936). The use of multiple measurements in taxonomic problems. *Annual Eugenics* **7**, II:179-188.

[4] Fodor J., Yager R., Rybalov A. (1997). Structure of Uninorms. *Int’l Jr. of Uncertainty, Fuzziness and Knowledge-based Systems*, 411-427.

[5] Han J., Moraga C. (1995). The influence of the sigmoid function parameters on the speed of backpropagation learning. In *From Natural to artificial Neural Computation* (J. Mira, Ed.), *LNCS* **930**, 195-201.

[6] Han J., Moraga C. (1996). Parametric Feedforward Network based Adaptive Fuzzy Modeling. *Proc. Int. Symp. Intelligent Industrial Automation and Soft Computing*, Reading UK, ICSC Academic Press, B–159–165.

[7] Han J., Moraga C., Sinne S. (1996). Optimization of feedforward neural networks. *Engineering Applications of Artificial Intelligence.* **9** (2), 109–119.

[8] Horikawa S.I., Furuhashi T., Uchikawa Y. (1992). A new type of Fuzzy Neural Network for Linguistic Fuzzy Modeling. *Proc. 2nd. Int. Conference on Fuzzy Logic and Neural Networks.* Iizuka, Japan, 1053–1056.

[9] Hornik K., Stinchcombe M., White H. (1989). Multilayer Feedforward Networks are universal approximators. *Neural Networks* **2**, 359–366.

[10] Jang J.S.R. (1993). ANFIS: Adaptive Network based Fuzzy Inference System. *IEEE Trans. on Systems, Man and Cybernetics* **23**, (3), 665–685.

[11] Keller J.M., Tager R.R., Tahani H. (1992). Neural Network implementation of fuzzy logic. *Fuzzy Sets and Systems* **45**, (1), 1–12

[12] Klement P., Mesiar R., Pap E. (1996). On the relationship of associative compensative operators to triangular norms and conorms. *Int’l Jr. of Uncertainty, Fuzziness and Knowledge-based Systems* **4**, (2), 129-144.

[13] Kosko B. (1994). Fuzzy systems as universal approximators. *IEEE Trans. Computers* **43**, (11), 1324–1333,

[14] Mesiar R., Komorniková M. (1998). Triangular norm-based aggregation of evidence under fuzziness. In: *Aggregation and Fusion of Imperfect Information* (B. Bouchon Ed.) *Studies in Fuzziness and Soft Computing* **12**, Physica Verlag, Springer.

[15] Moraga C. (1997). Properties of Parametric Feedforward Networks. *Proceedings XXIII Conferencia Latinoamericana de Informática*, Valparaíso, Chile, 861–870.

[16] Poggio T., Girosi F. (1990). Networks for approximation and learning. *Proceedings of the IEEE* **78**, 1481–1497.

[17] Silvert W. (1979). Symmetric Summation: A class of operations on fuzzy sets. *IEEE Trans. SMC* **9**, 659-667.

[18] Takagi H., Hagashi I. (1991). NN-driven fuzzy reasoning. *Int. Journal of Approximate Reasoning* **5**; (3), 191–212.

[19] Takagi H., Susuki N., Koda T., Kojima Y. (1992). Neural Networks designed on approximate reasoning architecture and their applications. *IEEE Trans. Neural Networks* **3**, (5), 752–760.

[20] Yi H.J., Oh K.W. (1992). Neural Network based Fuzzy Production Rule Generation and its application to an Approximate Reasoning Approach. *Proceedings 2nd. Int. Conference on Fuzzy Logic and Neural Networks*, Iizuka, Japan, 333–336.