ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 135-140

On the error function of Interval Arithmetic Backpropagation

Mercedes Fernández †, Carlos Hernández †.

† Department of Computers.
Jaume – I University.
Campus Penyeta Roja.
12071 CASTELLON.
SPAIN.
E-mail: espinosa@inf.uji.es

Abstract. A generalization of Multilayer Feedforward and Backpropagation to interval arithmetic was proposed several years ago. This generalization has several applications like the codification of expert's knowledge in the form of rules, "don't care attributes", missing inputs, etc. In the bibliography there are two error function and two training algorithms proposed, and in this paper we present the first comparison between both training algorithms. We have used a total number of 35 different problems for the comparison, Finally, the results are that one of the two training algorithms clearly out-perform the other.

1. Introduction.

Several years ago it was proposed a generalization of Multilayer Feedforward and Backpropagation to interval arithmetic [1], [2], [3]. This generalization allows to mixture interval inputs and point inputs (normal samples) inside the training set.
Since that, there are several general applications of this generalization in the bibliography. For instance, in the papers [1], [2] it is reported its usefulness to integrate expert's knowledge in the form of rules and samples in the training set, the samples are codified as usually and the rules can be codified by an interval. It is also reported its ability to handle "don't care attributes" in a simple and natural way, a "don't care attribute" is codified by an interval whose lower and upper limits include the range of variation of the input.
In [3], [4] it is reported its capability to codify missing or unknown inputs in the training and test set. An unknown input is represented by the interval [0,1] if the set of possible values of the attribute is [0,1]. It is shown that the network is able to respond to missing inputs in the test set if it is trained with missing inputs.
In [5] it is shown how to decrease the effect of weight errors by using interval arithmetic. It is proposed a training procedure based on interval arithmetic which decreases the output sensitivity to errors in the weights. This procedure has applications in digital and analog feedforward networks implementations.
Finally, in [6] a new approach to interval regression in neural networks is proposed.
However, there is still a problem in interval arithmetic Backpropagation. The problem is that two error function were proposed [4], [1] and there is not any comparison between the performance of both error function and their respective training algorithms. This paper will fill this gap by showing a complete comparison between both error functions.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 135-140

2. Two error functions.

As already told in the introduction, there are two error functions for interval arithmetic Backpropagation. The first one proposed by Ishibuchi *et al.* is the following:

$$Error = \sum_{k=1}^{Number\ of\ Ouputs} max\{(t_{Pk} - o_{Pk})^2 \ \ \forall o_{Pk} \in O_{Pk}\}$$

where o_{Pk} is a real number and O_{PK} is the interval output from the *k*th output unit. A learning algorithm can be derived from this equation in a similar way to Backpropagation taking into account the properties of interval arithmetic [2]. We will call this error function "Cost function I" in the rest of the paper.
The second error function was proposed by one of the authors of this paper [1]:

$$Error = \sum_{k=1}^{Number\ of\ Outputs} (t_{Pk} - o_{Pk}^L)^2 / 2 + \sum_{k=1}^{Number\ of\ Outputs} (t_{Pk} - o_{Pk}^U)^2 / 2$$

where o_{Pk}^L and o_{Pk}^U are the lower and upper limits of the interval ouput O_{Pk}, respectively. Another different training algorithm can be derived from this error function. We will call this error function "Cost function II".
There was a rather small comparison between both training algorithm [4], but the comparison was performed by using only one problem and it is absolutely insufficient. In this paper we present a more in deep comparison.
In order to define the percentage correct, several definitions of inequality can be used [4] in interval arithmetic. These definitions are:

$$Definition\ 1: \ O_{qh} <_1 O_{qk} \Leftrightarrow o_{qh}^U < o_{qh}^U$$

$$Definition\ 2: \ O_{qh} <_2 O_{qk} \Leftrightarrow o_{qh}^L < o_{qk}^L$$

$$Definition\ 3: \ O_{qh} <_3 O_{qk} \Leftrightarrow o_{qh}^U < o_{qk}^U \ \ and \ \ o_{qh}^L < o_{qk}^L$$

$$Definition\ 4: \ O_{qh} <_4 O_{qk} \Leftrightarrow o_{qh}^U < o_{qk}^L$$

And the definition of classification rule is the following:

$$O_{qh} <_i O_{qk} \ \ for \ \ h = 1,2,\ldots,C \ (h \neq k)$$

where $O_q = (O_{q1}, o_{q2}, \ldots, O_{qC})$ is the interval output vector from the trained neural network for the interval input vector X_q, if the relation holds we assign X_q to the class *k*.
We have used in our experiments the four classification rules, and the results are similar for every one. However, we will present here the results for classification rule number one (the first definition of inequality) because of the lack of space.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 135-140

3. Experimental results.

As we pointed out in the introduction, interval arithmetic codification can be used to codify missing inputs. So we have generated several missing inputs problems in order to compare both training algorithms.
We have selected seven problems from the UCI repository of machine learning databases, these problem are:

Credit Approval (CA): This problem concerns card applications. It has 15 nominal and continuos attributes, 2 classes, 453 training instances and 200 test instances.

Pima Indians Diabetes (PI): This problem has 8 attributes, 2 classes, 518 training instances and 250 test instances.

The Monk's Problems (MO1, MO2, MO3): We have used the three monk's problems. These problems were the basis of the first international comparison of learning algorithms. They are three problems with six attributes and two classes, 332 training instances and 100 test instances.

Display 1 (D1): This problem contains seven attributes, the seven segments of a light-emitting LED display, and 10 classes, the set of decimal digits. Each attribute value has a 10% probability of having its value inverted.

Display 2 (D2): It is the D1 problem, but additional seven irrelevant attributes are added to the instance space. It has 900 training instances and 2000 test instances.

And for each problem we have generated five new problems by randomly introducing missing inputs in the training and test set with percentages 5%, 10%, 20%, 30%, 40% over the total number of inputs. So, the number of different problems we have used for the comparison is 35.
In the following page we have the results for both training algorithms. For each problem, we have trained six different networks with different initialization, and we have averaged the results and obtained and error.
From the results, we can see that "Cost function II" proposed by Hernandez et al. [1], [2], clearly out-perform "Cost function I". The results of "Cost function II" are better than "Cost function I" in 21 of the 35 problems we have used. In 11 problems there is no difference within the errors between both training algorithms and in only 3 cases, "Cost function I" slightly improves the results of Cost function II".
We can also see, that the differences between both training algorithms increases as the number of training intervals in the training set increases. When the number of samples which are intervals increases "Cost function II" out-performs the results of "Cost function I". For instance, in the cases of 30% and 40% in the problem CA, in the cases of 20%-40% in the problem MO1 and MO3 and in the cases of 10%-40% in the problems D1 and D2.
So, we can finally conclude that the training algorithm derived from "Cost function II" is better than the other training algorithm from "Cost function I".

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 135-140

Table 1. Experimental Results.

	Percentage Correct. Cost function I (Ishibuchi *et al.*)				
Problem	5 %	10 %	20%	30%	40%
CA	85.24±0.11	83.7±0.2	84.10±0.01	77.25±0.01	52.20±0.01
PI	74.2±0.15	65.20±0.01	65.40±0.01	66.40±0.01	65.95±0.01
MO1	79.05±0.01	73.5±1.5	49.80±0.01	47.45±0.01	49.35±0.01
MO2	69.25±0.01	68.45±0.01	69.6±0.01	67.6±0.01	67.40±0.01
MO3	80.55±0.01	87.45±0.01	81.5±1.1	55.65±0.01	53.90±0.01
D1	71.4±0.5	54±9	8,70±0.01	11.25±0.01	9,30±0.01
D2	69.58±0.13	27±9	11.15±0.01	10.30±0.01	8.65±0.01

	Percentage Correct. Cost function II (Hernandez *et al.*)				
Problem	5%	10%	20%	30%	40%
CA	86.00±0.11	84.0±0.3	84.10±0.01	82.30±0.01	73.25±0.01
PI	69.0±0.4	72.8±0.4	65.82±0.03	66.40±0.01	65.95±0.01
MO1	79.067±0.017	68.4±0.3	73.7±0.6	70.35±0.01	67.5±0.1
MO2	69.25±0.01	68.45±0.01	69,60±0.01	67.60±0.01	67.40±0.01
MO3	82.5±0.3	88.0±0.4	85.2±0.6	78.2±0.2	77.75±0.01
D1	71.9±0.2	67.10±0.17	56±2	49.3±0.6	36.7±0.9
D2	68.78±0.12	62.94±0.07	57.1±1.3	49.9±0.6	35.5±0.8

4. Conclusions.

We have presented the two existing error functions and training algorithms for interval arithmetic Backpropagation.
And we have performed a comparison between both training algorithms. For the comparison we have used 35 different problems, the problems are from the UCI repository of machine learning databases and we have included randomly missing inputs in the problems.
The results show that there is one training algorithm proposed by Hernandez *et al.* [1], [2], the one derived from "Cost function II", which clearly out-perform the other. The difference between both training algorithms increases as the number of interval samples increases in the training set.

References.

1. Hernández, C.A.; Espí, J.; Nakayama, K., "A generalization of Backpropagation to Interval Arithmetic", World Congress on Neural Networks, (WCNN'93), vol. 4, pp. 131-134, 1993.

2. Hernández, C.A.; Espí, J.; Nakayama, K.; Fernández M., "Interval Arithmetic Backpropagation", Proceedings of 1993 International Joint Conference on Neural Networks, vol. 1, pp. 375-378, 1993.

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 135-140

3. Ishibuchi, H.; Miyazaki A., Kwon., K.; Tanaka, H., "Learning from Imcomplete Training Data with Missing Values and Medical Application", Proceedings of 1993 International Joint Conference on Neural Networks, vol. 2, pp. 1871-1874, 1993.

4. Ishibuchi, H.; Miyazaki, A.; Tanaka, H., "Neural-Network-Based Diagnosis Systems for Incomplete Data with Missing Inputs", Proceedings of International Conference on Neural Networks, vol. 6 , 3457-3460, 1994.

5. Anguita, D.; Ridella, S.; Rovetta, S., "Limiting the Effects of Weight Errors in Feedforward Networks using Interval Arithmetic", Proceedings of the International Conference on Neural Networks, vol. 1, pp. 414-417, 1996.

6. Haung, L.; Zhang, B., "A Novel-Network-Related Approach for Regression Analysis with Interval Model", Proceedings of the International Conference on Neural Networks, vol. 1, pp. 611-616, 1996.