Multi-Valued Neuron with a periodic activation function – learning with negative examples

Drd.Ing. Lupea V.M*, Dr.Ing. Cernazanu-Glavan C.**
*"Politehnica" University/Dept. of Computer Science, Timisoara, Romania
**"Politehnica" University/Dept. of Computer Science, Timisoara, Romania
*valilupea@yahoo.com, **cosmin.cernazanu@ac.upt.ro

Abstract— **The efficiency of Multi-Value Neuron and Multi-value Neuron with a periodic activation function are well presented in literature. Using these types of neurons, highly non-linear problems can be solved using a single neuron and not a multi-layer Neural Network (NN). On the other hand, using negative example during learning was shown to increase the overall efficiency. This paper brings these concepts together.**

Keywords- Multi-Valued Neuron (MVN), periodic activation function, l-periodic, negative example, classifier.

I. INTRODUCTION

In the current paper the efficiency of using negative example in the learning process of Multi-valued Neuron with a periodic activation function (MVN-P) is considered.

The efficiency of MVN-P in solving highly non-linear problems is presented in [5], [6], [7] or [9]. In [8] the author indicates the advantages of using in the training set, of a specific percentage of negative examples. Using negative example can increase the learning efficiency and also avoid the situation of over confidence of the final results.

The structure of the paper is as follows: Section II will present general aspects of MVN-P, Section III will focus on the aspect of using negative examples during learning, Section IV will detail the materials and methods used for investigating the advantages of using negative examples for MVN-P learning, Section V will show some results obtained during investigation, and finally Section VI will draw the observations and conclusions for this paper.

II. MVN-P: GENERAL ASPECTS

MVN-P was first introduced in [5], [6] and [7] as a successor of MVN, described in detail in [1] and [3].

A specific periodic activation function for MVN-P described by (1) in presented in [5].

$$P_l(z) = j \bmod k \quad (1)$$

if, $2\pi j/m \le \arg(z) < 2\pi(j+1)/m$, j = 0,1,…,m-1; m = kl, $l \ge 2$.

Figure 1 illustrates the characteristic of MVN-P learning based on the periodic activation function described by (1) [5].

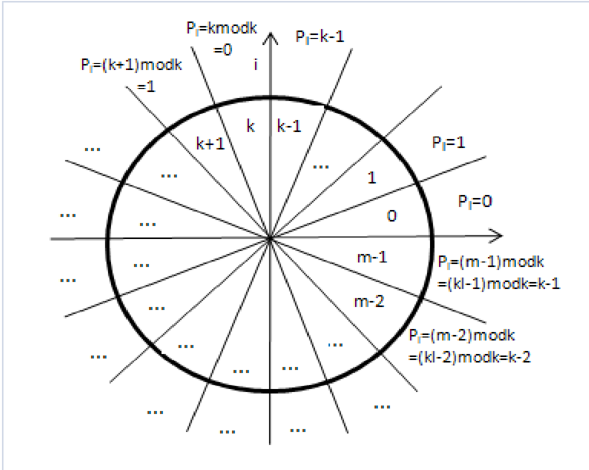

Figure 1. MVN-P learning characteristic based on l-periodic parameter

Correction rules for MVN-P weights are presented in [2] and [4]:

$$W_{r+1} = W_r + \frac{C_r}{(n+1)}(\varepsilon^q - \varepsilon^s)\overline{X} \quad (2)$$

$$W_{r+1} = W_r + \frac{C_r}{(n+1)|z_r|}(\varepsilon^q - \varepsilon^s)\overline{X} \quad (3)$$

Later, in [7], correction rules (2) and (3) were modified, resulting correction rules (4), respectively (5):

$$W_{r+1} = W_r + \frac{C_r}{(n+1)}(\varepsilon^q - \frac{z_r}{|z_r|})\overline{X} \quad (4)$$

$$W_{r+1} = W_r + \frac{C_r}{(n+1)|z_r|}(\varepsilon^q - \frac{z_r}{|z_r|})\overline{X} \quad (5)$$

In [7] and [9] various learning strategies for MVN-P are described in detail.

In [7], one of the learning strategies searches for the correct sector in the entire solution domain (m values). Two correct sectors are selected: one to the right and one to the left of the current sector. The decision between them is taken based on the angular distance; the closest one is chosen. The second strategy from [7] searches for the correct sector only in a sub-domain (k values) from the entire solution domain. The sub-domain chosen is the sub-domain in which the current sector is located.

In the learning strategy from [9] the correct sector is searched in the entire solution domain (m values) but the search is stopped after a given number the retries (set dynamically during learning) if the correct sector is not found; if this situation is reached, sector 0 is return as "correct" sector and the correction is done based on it [9].

III. LEARNING WITH NEGATIVE EXAMPLES

Using negative example during learning can improve the overall classification efficiency of a NN [8]. Nevertheless, the number of negative example used has a big influence over the system error. Too many negative examples will cause low recognition rates for the desired patterns [8] and too few negative examples will lead to a super confident NN that will not be able to recognize the undesired patterns [8].

In [8] the authors propose a mathematical model to establish the optimal number of negative example. Using a "regression analytic technique" [8] the authors identified that for a percentage of [10…20] % of negative example from the total number of all positive examples, the learning efficiency was increased [8].

Also, in [8], a general model for training with negative example was proposed and validated. The results showed that the model is valid and can identify with success the optimal number of negative example to be used [8].

IV. METHODS AND MATERIALS

For each output class, from the dataset, was used a MVN-P with the purpose of specializing it in detecting that specific class. All the other classes were considered to be negative example for this MVN-P. The goal was to obtain increased classification percentage for each class individually than for all classes considered together. Figure 2 illustrates the simplified system used for analysis.

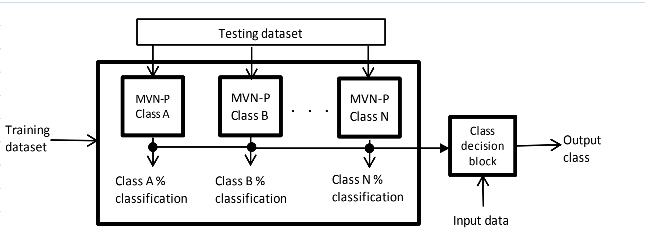

Figure 2. MVN-P classification system

In Figure 2 "MVN-P Class A" block refers to a MVN-P specialized in detecting output class A from a total of N classes. The efficiency (or probability) of MVN-P Class A block in detecting A output class is described by "Class A % classification" output parameter (e.g. a value of 98% means a 0.98 probability of correct class detection).

In Figure 3 an example of MVN-P block classifier is presented in more detail.

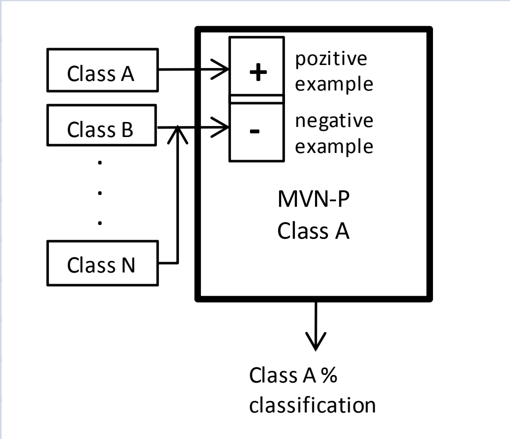

Figure 3. MVN-P block classifier

Output Class A is presented to the MVN-P as a positive example, while Class B through Class N are group together and presented as negative example. A percentage (%) detection value of Class A is outputted for MVN-P Class A block classifier.

All classifier blocks follow the same principal and based on each block class detection probability the MVN-P classification system decides the final output class for the current input data.

During learning, for all MVN-P blocks, the learning strategy from [9] was used, with weights correction rule described by (2) and periodic parameter l = 10.

V. RESULTS AND DISCUSSIONS

During testing various datasets were used and the results are presented in the following sub-chapters. For all datasets considered the data was divided into a training dataset (2/3 of the entire dataset) and a testing dataset (1/3 of the entire dataset). The training set is used for specializing MVN-P in specific output class detection and the testing set is used to determine the probability of correct class detection. Seven (7) independent runs were performed for all datasets, each using a different (random) training, respectively testing set.

A. *IRIS dataset*

IRIS dataset is a well known dataset composed of 150 instances grouped in 3 output classes (Iris Setosa, Iris Versicolour and Iris Virginica) each composed of 50 instances. Complete information and dataset download are available at [10].

During learning/testing 4 sets of sets are used. In Table I are listed the coding used for each set in part.

TABLE I. TRAINING/TESTING DATASETS CODING

Class output name	Setosa	Versicolour	Virginica
Class value	0	1	2
Training/Testing set 1 coding	0	1	2
Training/Testing set 2 coding	0	1	1

Training/Testing set 3 coding	0	1	0
Training/Testing set 4 coding	0	0	1

Set 1 uses all output classes without distinguishing between them (no classification in positive or negative examples). For Training/Testing Set 2 to 4 the negative examples are grouped and coded using the same value (e.g. for Set 3 output class Setosa and Virginica are grouped as negative example and coded using value 0 and class Versicolor is the positive example and coded with value 1).

In Table II are listed the results obtained for IRIS using a single MVN-P for all output classes (Set 1), respectively the results for MVN-P block classifier for each output class (Set 2 through 4).

TABLE II. RESULTS FOR IRIS DATASET

Set no.	Target output class	MVN-P learning efficiency (%)	Detection probability for target output class (x100)
1	0, 1, 2	98.2857	96.5714
2	0	100	100
3	1	77.4285	76.8571
4	2	98.7142	96

From Table II it can be observed that for output class 0 the probability of correct detection is 1 (100). For output class 2 the probability is also high with a value of 0.96, while for output class 1 it is quite low around 0.77. As conclusion, combining all MVN-P classifier blocks, the final output class decision is improved compared with the probability of a single MVN-P without negative examples (~0.965). The low value of output class 1 detection probability is not significant when using a MVN-P classifier system. We can consider, by elimination, that an output class 1 is present if it is not identified as output class 0 or output class 2.

B. SEEDS dataset

SEEDS dataset is another used and known dataset composed of 210 instances grouped in 3 output classes (Kama, Rosa and Canadian), each composed of 70 instances. Complete information and dataset download are available at [10].

During learning/testing 4 sets of sets are used. In Table III are listed the coding used for each set in part.

TABLE III. TRAINING/TESTING DATASETS CODING

Class output name	Kama	Rosa	Canadian
Class value	0	1	2
Training/Testing set 1 coding	0	1	2
Training/Testing set 2 coding	0	1	1
Training/Testing set 3 coding	0	1	0
Training/Testing set 4 coding	0	0	1

The same coding principal used for IRIS dataset is used also for SEEDS dataset with the same meaning and characteristics.

In Table IV are listed the results obtained for SEEDS for all MVN-P classifier blocks and for 1 MVN-P for all output classes.

TABLE IV. RESULTS FOR SEEDS DATASET

Set no.	Target output class	MVN-P learning efficiency (%)	Detection probability for target output class (x100)
1	0, 1, 2	84.9998	77.3465
2	0	80.9184	78.9790
3	1	95.9184	93.0608
4	2	97.5508	93.2648

From Table IV it can be observed that for all output classes the individual probability of correct detection is higher than the overall probability obtained for 1 MVN-P that is not using negative examples during learning (~0.77). Using MVN-P classifier system composed of individual MVN-P classifier blocks the final output class decision can be improved. As for IRIS case, the low probability value for output class 0 (~0.789) can be compensated by the high probability of class 1 and class 2 detection (both ~ 0.93) using the elimination principal.

C. Vertebral Column dataset (3 classes)

Vertebral Column dataset is another dataset used and it is composed of 310 instances grouped as follows: 60 instances of class Hernia, 150 instances of class Spondylolisthesis and 100 instances of class Normal. Complete dataset description and download are available at [10].

During learning/testing 4 sets of sets are used. In Table V are listed the coding used for each set in part.

TABLE V. TRAINING/TESTING DATASETS CODING

Class output name	Hernia	Spondylolisthesis	Normal
Class value	0	1	2
Training/Testing set 1 coding	0	1	2
Training/Testing set 2 coding	0	1	1
Training/Testing set 3 coding	0	1	0
Training/Testing set 4 coding	0	0	1

Again the same coding principal used for IRIS and SEEDS datasets is used also for Vertebral Column dataset with the same meaning and characteristics.

In Table VI are listed the results obtained for Vertebral Column for all MVN-P classifier blocks and for 1 MVN-P for all output classes.

TABLE VI. RESULTS FOR VERTEBRAL COLUMN DATASET

Set no.	Target output class	MVN-P learning efficiency (%)	Detection probability for target output class (x100)
1	0, 1, 2	79.6808	77.1974
2	0	84.6740	81.7302
3	1	97.5034	97.8015
4	2	84.3271	82.0048

From Table VI it can be observed, like for SEEDS dataset that for all output classes the individual probability of correct detection is higher than the overall probability obtained for 1 MVN-P that is not using negative examples during learning (~0.77). The highest probability (~0.978) was obtained for output class 1 due to the fact that this class is the best represented in the dataset (biggest number of instances compared with other classes). Once more we can conclude that using a MVN-P classifier system composed of individual MVN-P classifier blocks the final output class decision can be improved. For this dataset the elimination principal will not lead to a clear separation between classes (in this case between class 0 and class 2) because of the close values of class probability detection (~0.82). Even so, the individuals MVN-P classifier blocks (for both class 0 and 2) will detect more accurate an output class than the MVN-P without negative examples.

D. ECOLI dataset

ECOLI dataset is composed of 336 instances grouped as follows: 143 instances of class cp, 77 instances of class im, 52 instances of class pp, 35 instances of class imU, 20 instances of class om, 5 instances of class omL and 2 instances of both class imL and imS. Complete dataset description and download are available at [10].

TABLE VII. TRAINING/TESTING DATASETS CODING

Class output name	cp	im	imS	imL	imU	om	omL	pp
Class value	0	1	2	3	4	5	6	7
Training/ Testing set 1 coding	0	1	2	3	4	5	6	7
Training/ Testing set 2 coding	0	1	1	1	1	1	1	1
Training/ Testing set 3 coding	0	1	0	0	0	0	0	0
Training/ Testing set 4 coding	0	0	1	0	0	0	0	0
Training/ Testing set 5 coding	0	0	0	1	0	0	0	0
Training/ Testing set 6 coding	0	0	0	0	1	0	0	0
Training/ Testing set 7 coding	0	0	0	0	0	1	0	0
Training/ Testing set 8 coding	0	0	0	0	0	0	1	0
Training/ Testing set 9 coding	0	0	0	0	0	0	0	1

The same coding principal used for all previous datasets is used also for ECOLI dataset with the same meaning and characteristics.

In Table VIII are listed the results obtained for ECOLI for all MVN-P classifier blocks and for 1 MVN-P for all output classes.

TABLE VIII. RESULTS FOR ECOLI DATASET

Set no.	Target output class	MVN-P learning efficiency (%)	Detection probability for target output class (x100)
1	0, 1, 2…7	63.7755	60.8414
2	0	97.0025	93.7497
3	1	91.9641	88.2648
4	2	99.5538	99.4897
5	3	100	98.8518
6	4	93.5587	90.5607
7	5	99.0435	97.9588
8	6	100	99.6172
9	7	92.6022	89.7957

From Table VIII it can be observed again that using negative example during learning can improve classification probability of correct output class detection. Highest probabilities were obtained for output class 2, output class 3 and output class 6 (~0.99) due to the fact that they consist of the smallest instances numbers from all output classes; due to this fact MVN-P block classifiers specialized in output class 2, 3 and 6 detection can be used first to filter these three classes. With this re-arrange of MVN-P blocks the classification problem is reduced to a 5 class problem. Following the same principal of elimination as for the datasets presented above, and based on each MVN-P block individual detection probability, the “Class detection block" can detect more accurate an output class than the MVN-P without negative examples.

VI. CONCLUSIONS

In this paper the concept of learning with negative examples was applied for MVN-P. It was proven in literature that using a certain percentage of negative examples from the total number of positive examples can increase the classification accuracy of a NN [8]. A MVN-P classifier system was proposed that consists in N (N is the number of output classes) individual MVN-P block classifiers. Each MVN-P block classifier is specialized in identifying a specific output class. In order to fulfill this goal all training instances of one specific class were presented to the MVN-P classifier block as positive example; all other instances were grouped and presented as negative example. Various datasets were used (IRIS, SEEDS, Vertebral Column and ECOLI) and in all cases the usage of negative examples increased the overall probability value of correct output class detection.

In conclusion, using negative examples during learning process can increase the classification accuracy also for the case of MVN-P. As future work, it will be interesting to modify the MVN-P classifier system for the situation when the detection probability of 2 classes is close and the elimination principal cannot be applied.

ACKNOWLEDGMENT

This work was partially supported by the strategic grant POSDRU/CPP107/DMI1.5/S/77265 (2010) of the Ministry of Labor, Family and Social Protection, Romania, co-financed by the European Social Fund – Investing in people.

REFERENCES

[1] Aizenberg N.N. and Aizenberg I.N , "CNN based on multi-valued neuron as a model of associative memory for gray-scale images", Proc. of the 2-d IEEE International Workshop on Cellular Neural Networks and their Applications, Munich, pp. 36-41, October 12-14 (1992).

[2] I. Aizenberg, N. Aizenberg, and J. Vandewalle, Multi-Valued and Universal Binary Neurons: Theory, Learning, Applications. Boston, MA: Kluwer, 2000.

[3] Igor Aizenberg and Claudio Moraga, "Multi-Layered Neural Network Based on Multi-Valued Neurons (MLMVN) and a Backpropagation Learning Algorithm", Technical Report ISSN, pp. 1433-3325, April 2004.

[4] I. Aizenberg and C. Moraga, "Multilayer feedforward neural network based on multi-valued neurons (MLMVN) and a backpropagation learning algorithm," Soft Comput., vol. 11, no. 2, pp. 169–183, January 2007.

[5] Igor Aizenberg, "A multi-valued neuron with a periodic activation function", International Joint Conference on Computational Intelligence, IJCCI 2009

[6] Igor Aizenberg, Matthew Caudill, Jacob Jackson, and Shane Alexander, "Learning Nonlinearly Separable mod k Addition Problem Using a Single Multi-Valued Neuron With a Periodic Activation Function", WCCI 2010 IEEE World Congress on Computational Intelligence, pp. 18-23, Barcelona, Spain, July 2010.

[7] Igor Aizenberg, "Periodic Activation Function and a Modified Learning Algorithm for the Multi-valued Neuron", IEEE - IEEE Transaction on neural networks, Vol. 21, No. 12, December 2010.

[8] Cernazanu-Glavan, Cosmin, and Stefan Holban. "A Model for Determining the Number of Negative Examples used in Training a MLP." Innovations in Computing Sciences and Software Engineering. Springer Netherlands, 2010. 537-542.

[9] Lupea, Valentin Mircea, "Multi-Valued Neuron with a periodic activation function — New learning strategy", 2012 IEEE 8th International Conference on Intelligent Computer Communication and Processing, 2012, pp. 79-82, Cluj-Napoca, Romania, August 2012.

[10] A. Asuncion and D. J. Newman. (2007). UCI Machine Learning Repository. School Inform. Comput. Sci., Univ. California, Irvine [Online]. Available: http://www.ics.uci.edu/ mlearn/MLRepository.html