The Adaptive Weight Using RAM

Eduardo do Valle Simões†, Luís Felipe Uebel†, Yuzo Ueno & Dante Augusto Couto Barone

Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre - RS - Brazil
edvs1@ukc.ac.uk; lfu20@eng.cam.ac.uk; yuzo@inf.ufrgs.br; barone@inf.ufrgs.br

ABSTRACT

This article analyses the saturation problem of a RAM neural network, a n-tuple classifier [1] containing 340 12-input neurons applied to the character recognition task, using the British mail data bank [2]. It presents data to evaluate this problem and correlates it to other characteristics of the RAM nets [3] [4]. Therefore, two novel approaches were suggested to reduce the network saturation and improve the recognition level: the Filtered RAM and the Adaptive Weight Using RAM (AWURAM). The first version simply multiplies each input vector by a digital filter during the training and the recall phases. The second approach associates the weight concept to the network in order to distinguish different regions among the trained classes.

1. INTRODUCTION

Boolean Artificial Neural Networks have been used in many situations that need fast training, processing speed, and easy hardware implementation [1-6]. Despite having all of these advantages, the n-tuple classifier [1], called here "the RAM model", has presented saturation problems for large training pattern sets [4]. In applications such as the character recognition task, using the British mail database [4, 6], containing 300 samples of numeric digits from "0" to "9", this network can present a low recognition level. Another characteristic that should be considered when dimensioning a RAM neural network architecture is pointed out here. It was named "the sureness level" and shows how accurate is the network response [6]. In other words, the sureness level indicates if the number of activated neurons in the wining class is significantly higher than the number of activated neurons in the other classes (i.e. the network response can be correct, but noisy input patterns could cause errors).

The sureness level was defined in this paper as the percentage of the recognised samples where the sum of activated neurons of the wining class is at least 20% greater than the sum of the second wining class.

A neural network architecture containing 340 12-input neurons are applied to cover all 384 pixels of a 24 x 16 character grid of the British Mail database. This architecture are used to test the performance of the suggested techniques, that present a different strategy to deal with the saturation problem of the RAM neural networks.

2. THE FILTERED APPROACH

Considering large training sets [11], it may occur that the majority of the grid pixels are set to on during the training phase. It happens when the training set has a good diversity of input patterns per class and can lead to a process where the logic value "1" is written in almost all memory positions of the neurons [4, 6]. Therefore, the same memory positions of neurons of different classes are addressed and written by noisy patterns. In that case, two digits of different classes can be confused by the classifier after the training process of many input patterns, as it can be seen in figure 1. Here, after the overlapping of 300 training patterns, the class of the digit "1" were trained with so many patterns that even an input pattern of the digit "0" could have generated some of the pixels. This process can conduct to the saturation of the neuron memories and decrease the recognition level [4].

In order to face this problem, a filter was built to detect the most significant pixels of the character from the scanning process noise and present these pixels to the network in the training process. A different filter was implemented for each class according to the following process:

1. All available training pattern for the specific class were overlapped into a grid;
2. Each grid pixel has an accumulator to calculate how many times it has been turned on;
3. Only the pixels with a sum bigger than a threshold level (20% of the higher value, for example) are set on ("1") in the filter, the others are turned off ("0").

---

† Ph.D. Students Granted by CNPq - Brazil.

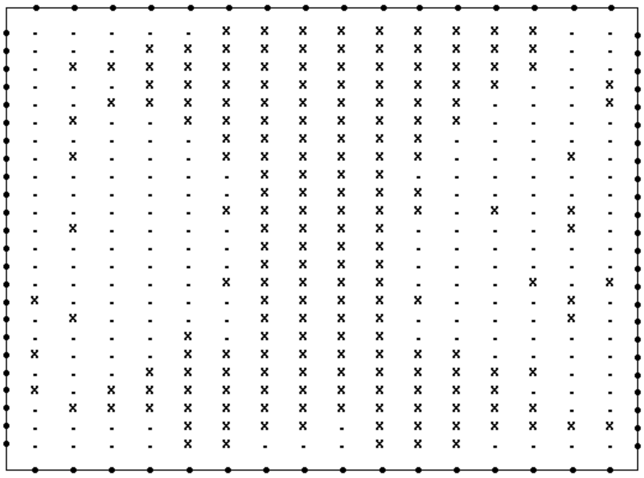

Figure 1a - Overlap of 100 samples

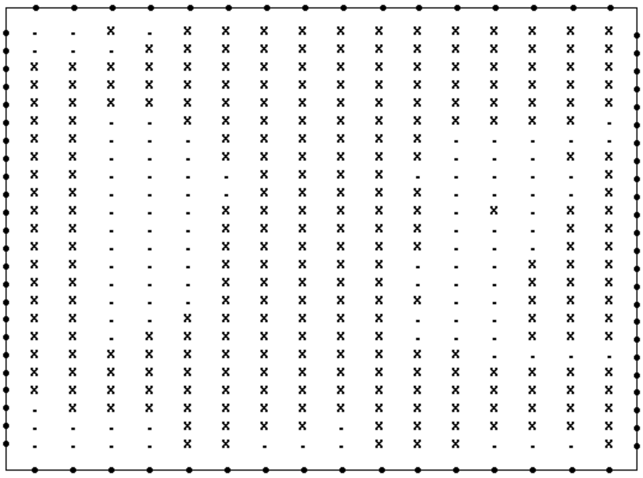

Figure 1b - Overlap of 300 samples

Therefore, each class will have a specific filter grid, where the “1” values represent the most significant pixels of the class. Figure 2 represents how a filter was built with 300 training patterns for the class of the digit “4”. This process is repeated for all classes before the network training phase. During the training phase, each input pattern is multiplied by the filter of the class it belongs before it is presented to train the neural network. This process reduces the network saturation once only the most relevant information of each pattern is stored in the neuron memory [7-8]. The discarded pixels do not degrade system performance for they result from a noisy data acquisition process.

A further modification was applied in the original model once the results of the tests showed that it was possible to make use of the same filter in the recall phase. Therefore, when a new input pattern is been tested by a specific class in the network, it is firstly multiplied by the filter of that class, built in the training phase. Hence, only the most relevant pixels are presented to the neurons of the each class. Then the class with the highest number of activated neurons are chosen as the winner.

Section 4 presents the obtained results with the filtered model, where the improvements in the recognition level and the sureness level can be noticed.

3. THE WEIGHT USING APPROACH

The filtered approach succeeded in reducing the influence of the noise introduced by the scanning process. In order to improve the recognition level it is necessary to have a strategy to deal with similarities among different classes during the recall phase [8-10]. The adaptive weight-using RAM (AWURAM) addresses this problem, detecting and privileging the most representative pixels of each class. This approach tries to decrease the importance of the pixels that one class has in common with the others.

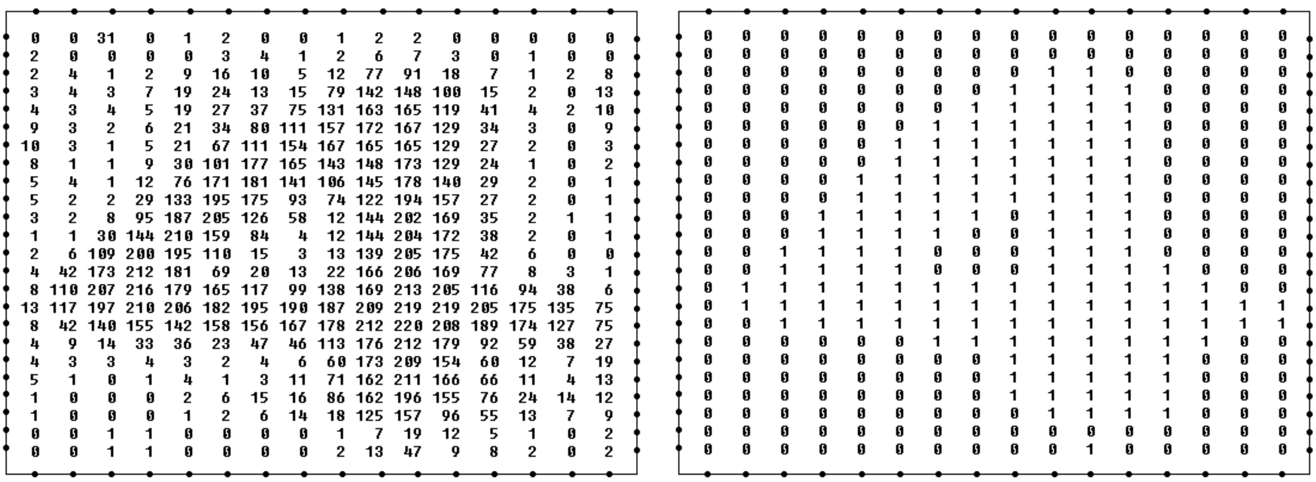

Figure 2 - The filter of the class “4” is built from the overlap of all available training samples of the digit “4”. The most frequent pixels are marked with “1”.

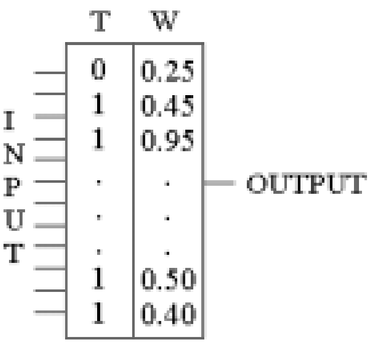

Figure 3 - A Weight-using RAM neuron: each memory position contains the training phase result (T) and its weight (W).

Figure 3 represents a weight-using neuron, where each input addresses a memory position containing the training phase result and its weight. The neuron output is then calculated by multiplying the selected memory position by its weight. The output of all neurons of each class is then summed to indicate what class is the winner.

The network is trained using the same filters of the previous model. The weights are initialised as “0” and can have any value between “0” and “1”. During the recall phase, the neurons connected to the same pixels of the grid (one neuron per class) are analysed at the same time. Then, it is verified how many of these neurons are activated (addressed content equal to “1”), and the weight of the selected memory positions is updated according to Table I. The number of activated neurons indicates the corrective factor to be summed to the weight of the selected memory positions. When 3 or more neurons of different classes, addressing the same pixels in the grid, are active, the weight updating process decreases the weight of the output of these neurons, trying to reduce the influence of this portion of pixels in the result.

In the weight-using model, common pixels of digits belonging to different classes are less important for choosing the winner class than distinct ones. Section 4 presents the advantages of this model over the others if the sureness level is considered.

Table I – Updating factors for the weight of the selected memory positions

Activated Neurons	Updating Factor
1	1
2	0.5
3	-0.1
4	-0.1
5	-0.1
6	-0.15
7	-0.15
8	-0.20
9	-0.20
10	-0.20

4. OBTAINED RESULTS

The British mail data bank, containing 300 samples of the digits “0” to “9” (3.000 samples), was used to evaluate the performance of the networks. Table II presents the obtained results. The first row shows the size of the training set (number of samples x 10 digits); the second presents the size of the recall set. The Error row shows the percentage of the recall patterns that were not recognised correctly; and the sureness level represents the quality of the results.

Table II shows that the recognition level tends to be improved in the original RAM when more samples are used to train the neural net, but the network saturation, shown by the decay in to the sureness level, also rises. When the filtered RAM is used, a significant increase of the recognition level can be noticed. The sureness level also indicates less saturation of the network, but it still reaches high values for a large training set. The weight-using RAM has the best performance when small training sets are used (100 - 200 samples), but shows a little loss of power for larger sets. The results show that the last model has a low saturation rate even for large training sets.

Table II – The recognition level and sureness level (SL) for the implemented networks

Training	Recall	Original RAM		Filtered RAM		Weight-using RAM	
Patterns	Patterns	Error	SL (> 20%)	Error	SL (> 20%)	Error	SL (> 20%)
10	2990	2700 (90.60%)	523 (17.50%)	893 (29.86%)	1842 (61.61%)	628 (21.00%)	2368 (79.20%)
50	2950	669 (22.67%)	1366 (46.31%)	476 (16.13%)	1837 (62.28%)	36 (1.22%)	2806 (95.04%)
100	2900	419 (14.45%)	1460 (50.35%)	112 (3.86%)	2088 (72.00%)	7 (0.24%)	2842 (98.00%)
200	2800	266 (9.50%)	1558 (55.65%)	37 (1.32%)	2104 (75.15%)	0 (0.00%)	2789 (99.65%)
500	2500	141 (5.64%)	1180 (47.20%)	15 (0.60%)	1733 (69.32%)	6 (0.24%)	2492 (99.68%)
800	2200	108 (4.90%)	805 (36.6%)	19 (0.86%)	1443 (65.59%)	5 (0.23%)	2196 (99.82%)
1000	2000	90 (4.50%)	599 (29.95%)	13 (0.65%)	1235 (61.75%)	6 (0.30%)	1997 (99.58%)
1280	1720	75 (4.36%)	413 (24.01%)	10 (0.58%)	1019 (59.30%)	5 (0.29%)	1719 (99.94%)
1500	1500	66 (4.40%)	313 (20.87%)	5 (0.33%)	839 (55.94%)	8 (0.53%)	1499 (99.94%)
2000	1000	52 (5.20%)	138 (13.8%)	4 (0.40%)	546 (54.60%)	7 (0.70%)	995 (99.50%)

Table III – Processing time of the implemented networks recall phase (2.000 samples)

Network	Time (min:sec)
Original RAM	2:15
Filtered RAM	3:44
Weight-using RAM	5:33

The sureness level for the AWURAM, opposite to the other models, seems to rise when more training patterns are used, yet for more than 1.000 training samples a larger neural architecture should be chosen.

The proposed modifications presented good results related to the recognition level and the sureness level, but all these improvements resulted in a reduction of the system performance, as shown in Table III. All models where implemented in C language, using a SUN C.3.0.1 Compiler and the SUN OS5.5 in a SUN Sparc Station 20, with 256Mb of memory.

5. CONCLUSION

The main difference between the Filtered RAM and the AWURAM is that the first has implicit neuron weights equal to 0 or 1, and the other has explicit neuron weights among 0 and 1. The most important feature of the new AWURAM model is the capability of changing and updating its neuron weights when it faces unknown patterns. Thereby, the AWURAM model can reorganise its internal representation of the numeric digits from 0 to 9. In this way, AWURAM presents adaptable features for a limited environment.

The Filtered RAM, the AWURAM model and the original RAM were trained and tested using the same numeric digits from the British mail data bank. The obtained results have shown the advantages of these new techniques: the filtered RAM presented an error of 1.32% and the weight-using RAM reaches 0.01%, in comparison to 9.50% of the original RAM model, considering 200 training vectors. The disadvantage of both developed techniques is the network processing time, that is greater than the original RAM model. This strategy proved to be a useful approach to deal with the saturation problem faced in previous work, that applied the fast training and simple hardware design of the RAM model in to mobile robot control [5] and pattern recognition [6] tasks.

6. REFERENCES

[1] R. Rohwer and M. Morciniec, **The Theoretical and Experimental Status of the N-Tuple Classifier**, Technical Report, Dept. Computer Science and Applied Mathematics, Aston University, Birmingham, Uk, May 1995, 59 p.

[2] I. Aleksander, W. V. Thomas and P. A. Bowden, **WISARD, A Radical Step Forward in Image Recognition**, Sensor Review, 4, 1984, pp. 120-124.

[3] R. Rohwer and M. Mordiniec, **A Theorical and Experimental Account of N-Tuple Classifier Performance**, Neural Computation, 8, 1996, pp. 629-642.

[4] E. C. D. B. Filho, M. C. Fairhurst and D. L. Bisset, **Analysis of Saturation Problem in RAM-Based Neural Network.** Electronics Letters, Vol. 28, No. 4, February 1992, pp. 345-346.

[5] S. S. C. Botelho, E. V. Simões, L. F. Uebel and D. A. C. Barone, **High Speed Neural Control for Robot Navigation**, 1996 IEEE International Conference on Systems, Man and Cybernetics, Beijing, China, October 1996, pp. 421-429.

[6] E. V. Simões, L. F. Uebel and D. A. C. Barone, **Hardware Implementation of RAM Neural Networks**. Pattern Recognition Letters, n. 17, 1996, pp. 421-429.

[7] C. Myers, **Learning with Delayed Reinforcement Through Attention-driven Buffering**. Journal of Neural Systems, 1 (4), 1991, pp.337-346.

[8] D. Gorse and J. Taylor, **Training Strategies for Probabilistic RAMs**. Parallel Processing in Neural Systems and Computers, Elsevier Science, Amsterdam, 1990, pp. 161-164.

[9] E. V. Simões, L. F. Uebel and D. C. Barone, **Fast Prototyping of Artificial Neural Network: GSN Digital Implementation.** IV International Conference on Microelectronics for Neural Networks and Fuzzy System, Turin, Italy, September 1994, pp. 192-201.

[10] C. E. Myers, **Delay Learning in Artificial neural Networks.** Chapman & Hall, London, 1992, 157 p.

[11] D. Gorse e J. Taylor, **Training Strategies for Probabilistic RAMs**. Parallel Processing in Neural Systems and Computers, Elsevier Science, Amsterdam, 1990, pp. 161-164.