COMPLEXITY REDUCTION IN NEURAL NETWORKS APPLIED TO TRAFFIC SIGN RECOGNITION TASKS

R. Vicen-Bueno, R. Gil-Pita, M.P. Jarabo-Amores and F. López-Ferreras

Departamento de Teoría de la Señal y Comunicaciones
Escuela Politécnica Superior, Universidad de Alcalá
Ctra. Madrid-Barcelona, km. 33.600, 28805, Alcalá de Henares - Madrid (SPAIN)
E-mails:{raul.vicen, roberto.gil, mpilar.jarabo, francisco.lopez}@uah.es

ABSTRACT

This paper deals with the application of Neural Networks (NNs) to the problem of Traffic Sign Recognition (TSR). The NN chosen to implement the TSR system is the Multilayer Perceptron (MLP). Two ways to reduce the computational cost in order to facilitate the real time implementation are proposed. The first one reduces the number of MLP inputs by pre-processing the traffic sign image (blob). Important information is kept during this operation and only the redundancy contained in the blob is removed. The second one looks for neural networks with reduced complexity by selecting a suitable error criterion for training. Two error criteria are studied: the Least Square error (LS) and the Kullback-Leibler error criteria. The best results are obtained using the Kullback-Leibler error criterion.

1. INTRODUCTION

Systems dedicated to Traffic Sign Recognition (TSR) usually have two specific stages. The fist one is related to the detection of traffic signs in a video sequence or image. The second one is related to the recognition of these detected signs, which is paid special attention in this work. The performance of these stages highly depends on lighting conditions of the scene and the state of the road sign due to deterioration or vandalism. Another problem to surpass is the rotation, translation or inclination of the sign. Its perfect position is perpendicular to the trajectory of the vehicle, however many times it is not like that. Problems related to the traffic sign size are of special interest too. Although traffic sign sizes are normalized, we can find signs of different ones. So, the recognition of a traffic sign in this environment is not easy.

The TSR problem has been studied many times in the literature. The works [1, 2, 3] solve this problem using the correlation between the traffic sign and the elements of a data base. This technique involves great computational cost. In other works [4], Matching Pursuit (MP) is used in two stages: training and testing. The training process finds a set of best MP filters for each road sign. The testing process projects the input unknown road sign to different MP filters to find the best match. This method also implies high computational cost, specially when the number of elements grows up. In a recent work [5], a Neural Network (NN) following the Adaptive Resonance Theory is used as classification technique. This work applies this technique to the whole image, where many traffic signs can exist. This involves that the NN complexity must be very high in order to recognize all the signs contained in the image.

On the other hand, many works are applied in a single-frame way. For example, in [6] a Kalman filter is used to track a sign over the frames until it is sufficiently large to be recognized as a specific standard sign. A recent work [7] presents an automatic road sign detection and recognition system based on a computational model of human visual recognition. This work gives good results because of the tracking system, although they consider road signs are always composed of a colour rim with a black/white interior. Its main drawback is the high computational cost needed to implement it.

This work is supported by Spanish MEC Project TEC2004/03511/TCM.

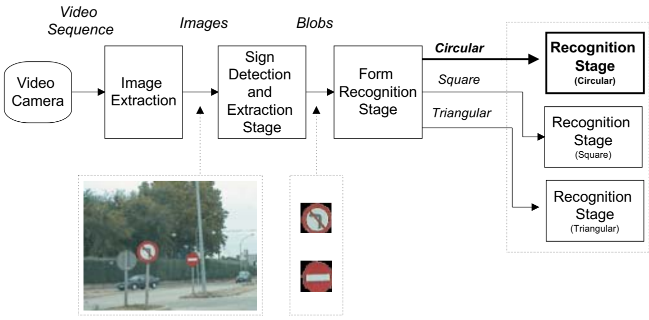

Figure 1: Blocks of the TSR System

The objective of this work is to present two different ways to reduce the computational cost. The first one applies a pre-processing of the traffic signs detected to reduce the number of inputs of the Multilayer Perceptrons (MLP) used as classifiers. The second one looks for a good error criterion which provides good performance with small networks.

2. DESCRIPTION OF THE TRAFFIC SIGN RECOGNITION SYSTEM

The TSR system used is presented here. The traffic signs obtained with it and considered for the experiments are described, too.

The blocks the TSR system is composed of are shown in Figure 1. The *Video Camera* takes a *video sequence*. The *Image Extraction* block is the responsible for creating *images*. The *Sign Detection and Extraction Stage* extracts all the traffic signs contained in each image and generates the small images called *blobs*, one per sign. Figure 1 also shows an example of the way this block works. The *Form Recognition Stage* is the responsible for discerning among the different forms: circular, square, triangular and others. Once the blob form is classified, the *Recognition Stage* has the responsibility to recognize which is the exact type of signal. This stage is divided in two parts: *Traffic Sign Pre-processing Stage* and *Recognition Core*. This paper propose a feasible implementation of these two blocks, that provide both low complexity and high correct recognition probability.

2.1 Traffic Sign Pre-processing Stage

Each blob presented at the input of the recognition stage has the information of the red (R), green (G) and blue (B) colours. The blob dimension is $30x30$ pixels for each component (R, G and B). So, the total size of each blow is 2700 pixels. Due to the dimensions of the blob, the purpose of this stage is to reduce the redundancy of information given to the recognition core in order to reduce its computational cost.

Consider $\mathbf{B}$ is the matrix that contains the three colour components of the blob. Also, consider $\mathbf{B}'$ results from representing $\mathbf{B}$ in a grey scale. This change from colour to grey scale is calculated with (1). The values of $b_{i,j}$ and $b'_{i,j}$ are the elements of the *i-th* row and *j-th* column of the matrix $\mathbf{B}$ and $\mathbf{B}'$, respectively, where both indexes (i and j) varies from 1 to 30.

$$b'_{i,j} = 0.49 b_{i,j} + 0.29 b_{i+30,j} + 0.22 b_{i+60,j} \tag{1}$$

The averages normalized to the maximum pixel value (2^8) of R (MR), G (MG) and B (MB) are calculated with (2), (3) and (4), respectively.

$$MR = \frac{1}{256}\left(\frac{1}{900}\sum_{i=1}^{30}\sum_{j=1}^{30} b_{i,j}\right) \tag{2}$$

$$MG = \frac{1}{256}\left(\frac{1}{900}\sum_{i=31}^{60}\sum_{j=1}^{30} b_{i,j}\right) \tag{3}$$

$$MB = \frac{1}{256}\left(\frac{1}{900}\sum_{i=61}^{90}\sum_{j=1}^{30} b_{i,j}\right) \tag{4}$$

The normalized vertical ($\mathbf{vh}$) and horizontal ($\mathbf{hh}$) histograms are calculated with (5) and (6), respectively.

$$vh_i = \frac{1}{30}\sum_{j=1}^{30}\left(b'_{i,j} > T\right) \quad , \; i = 1,2,\ldots,30 \tag{5}$$

$$hh_j = \frac{1}{30}\sum_{i=1}^{30}\left(b'_{i,j} > T\right) \quad , \; j = 1,2,\ldots,30 \tag{6}$$

T is the adaptive threshold calculated with (7) for each blob. vh_i is the *i-th* value of $\mathbf{vh}$ and corresponds to the percentage of values of the *i-th* row that are greater than the threshold T. hh_j is the *j-th* value of $\mathbf{hh}$ and corresponds to the percentage of values of the *j-th* column that are greater than the threshold T.

$$T = \frac{1}{900}\sum_{i=1}^{30}\sum_{j=1}^{30} b'_{i,j} \tag{7}$$

This pre-processing provides an observation vector $\mathbf{x}$ of 63 elements. It is composed of: $\mathbf{vh}$ (30 elements), $\mathbf{hh}$ (30 elements), MR, MG and MB.

2.2 Traffic Sign Data Base Description

For the experiments presented in this paper, eight different types of circular traffic signs were considered. These signs belong to the international traffic code. Figure 2 (Normal Traffic Signs) shows the different classes of traffic signs considered in this work. These signs have been collected with the TSR system presented above. So, they present distortions due to the problems described in section 1 (deterioration, vandalism, lighting variation, etc.). Some examples are shown in Figure 2 (Traffic Signs with problems). The problems caused by vandalism are shown in the examples of classes S_1 and S_3. The problems related to the blob extraction in the *Sign Detection and Extraction Stage* (not a correct fit in the square image) are shown in the examples of classes S_1, S_3 and S_7. Examples of signs with problems of rotation, translation or inclination are those of classes S_3, S_4, S_5, S_7 and S_8. Finally, the differences of brightness are observed in both parts of Fig. 2.

The data base has been divided into three sets: train, validation and test. The first one is used to train the MLP. The second one is used as validation set during the MLP training to improve generalization [8]. And the last one is used to evaluate the performance of the trained MLP.

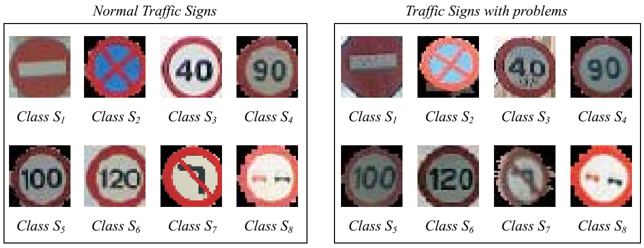

Figure 2: Eight different classes of international traffic signs (normal and with problems)

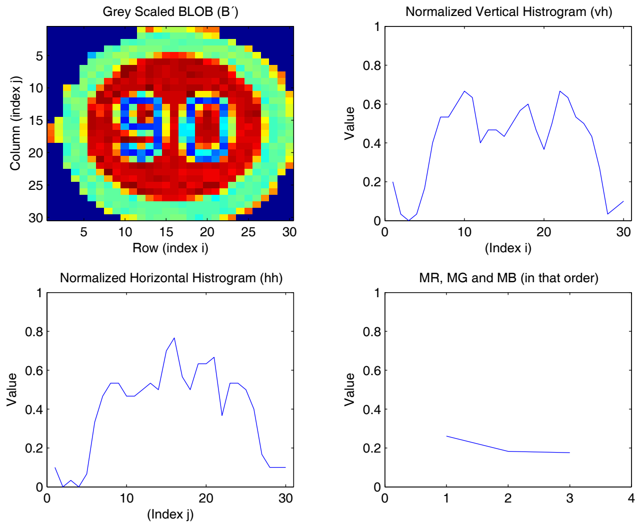

Figure 3: Elements of the observation vector $\mathbf{x}$ ($\mathbf{vh}$, $\mathbf{hh}$, MR, MG and MB) for an example blob of the class S_4

The total number of traffic signs (blobs) considered for the experiment is 235. So, after pre-processing each blob, a total number of 235 observation vectors of 63 samples length each is obtained. An example of the pre-processing applied to a given blob is shown in Figure 3. The size of the train, validation and test sets are 79, 78 and 78 observation vectors (patterns), respectively.

2.3 Recognition Core

Neural networks have been considered to implement the Recognition Core. In particular, Multilayer Perceptrons are considered. The Perceptron was developed by F. Rosenblatt [9] in the 1960s for optical character recognition. The Perceptron has multiple inputs fully connected to an output layer with multiple outputs. Each output y_j is the result of applying the linear combination of the inputs to a non linear function called activation function. MLPs extend the Perceptron by cascading one or more extra layers of processing elements. These extra layers are called hidden layers, since their elements are not connected directly to the external world. Figure 4 shows an MLP ($I/N/O$) with I inputs, one hidden layer with N neurons and O outputs. For the results presented in this paper, the hidden layer of the MLP is composed of neurons with log-sigmoidal activation functions:

$$L(\mathbf{z}) = \frac{1}{1 + exp(-\mathbf{z})} \tag{8}$$

The classifier performance can be specified with the *probability of correct classification* (P_{cc}) and the *probability of misclassification* (P_{mc}) for each hypothesis (class) or the *total correct rate* (P_c) and the *total error rate* (P_e) for all the hypotheses (classes). The

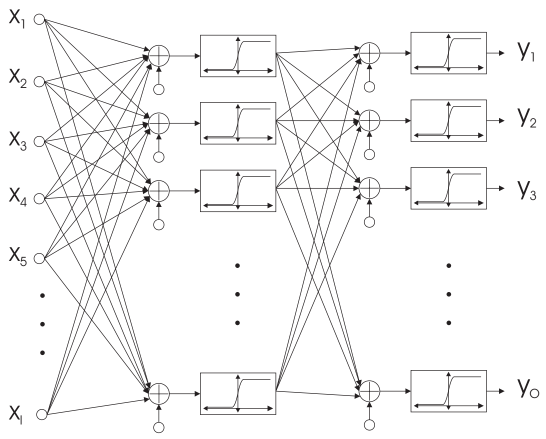

Figure 4: Structure of a MLP

P_{cc} is the probability of a pattern corresponding to a hypothesis is correctly classified and the P_{mc} is the probability that a pattern corresponding to a hypothesis is wrongly classified (P_{mc}=*1*-P_{cc}). The P_c and P_e express the percentage of total classification successes and errors (P_e=*1*-P_c) for all the hypothesis (classes), respectively. In this paper we present results of the P_e, and the overall number of errors made by a given system.

The *Back-Propagation Algorithm*[8][10] with cross-validation is used to train the MLP. This algorithm tries to find the minimum of the error surface given by the error criterion. Two error criteria are studied in this work: the Least Square (LS) and the Kullback-Leibler (KL) error criteria.

The use of the Kullback-Leibler error criterion gives rise to small networks with better or equal results than high networks trained using the Least Square error criterion, as can be seen in section 3. This fact makes possible the implementation of low complexity systems.

2.3.1 Least Square Error Criterion

The LS error criterion applied to train a MLP is given by (9).

$$E_{LS} = \frac{1}{M}\sum_{l=1}^{M}\sum_{c=1}^{C}[y_c(l) - d_c(l)]^2 = \frac{1}{M}\sum_{l=1}^{M}\sum_{c=1}^{C}[e_c(l)]^2 \qquad (9)$$

where M is the number of training patterns. $y_c(l)$ is the *c-th* element of the output vector for input pattern $\mathbf{x}(l)$. $d_c(l)$ is the *c-th* element of the desired output vector which is equal to 1 if the input belongs to class c, and equal to 0 if the input does not belong to class c. $e_c(l)$ is the difference between the *c-th* MLP output and its desired output when the *l-th* pattern is applied to the input.

It can be demonstrated that the LS error measure for MLPs is equivalent to the expected value of the square of the distance between its output and the posterior probability of class given the input [11]. This equivalence is derived for a two-class problem with a single output MLP. The results can be generalized to large problems. So, for a two-class problem, the LS measure is given by (10).

$$E_{LS} = \frac{1}{M}\sum_{l=1}^{M}(e(l))^2 \qquad (10)$$

If the number of training patterns is large, the summation in (10) converge to the expected value, so the error measure becomes

$$E_{LS} = E\{(y-d)^2\} \qquad (11)$$

$$E_{LS} = E\{(\mathbf{y} - E\{d|\mathbf{x}\})^2\} + E\{(E\{d|\mathbf{x}\} - d)^2\} \qquad (12)$$

For the desired output defined to be either 0 or 1, the conditional expectation in (12) converges to the posterior probability, i.e.

$$E\{d|\mathbf{x}\} = P\{c|\mathbf{x}\} \qquad (13)$$

E_{LS} now becomes

$$E_{LS} = E\{[y - P\{c|\mathbf{x}\}]^2\} + E\{[P\{c|\mathbf{x}\} - d]^2\} \qquad (14)$$

The first item in (14) is the only one which depends on the MLP parameters. Therefore, minimizing E_{LS} results in a LS estimate of the posterior probability of the class c given the input $\mathbf{x}$. The LS estimate tends to overemphasize the large probabilities and deemphasize the smaller ones. Therefore, the LS method will estimate the higher probabilities more accurately than the lower ones.

2.3.2 Kullback-Leibler Error Criterion

The use of feedforward neural nets produce reliable estimates of the posterior probability. Therefore, a distance measure which is sensitive to the difference between the true and estimated probability must be selected. One such criterion is the Kullback-Leibler information measure [12]. A modified version of the KL measure is proposed in [11]. They propose to minimize the following error measure (15) when estimating the parameters of the MLP.

$$E_{KL} = -\frac{1}{M}\sum_{l=1}^{M}\sum_{c=1}^{C}\frac{1}{2}log[y_c(l) - \bar{d_c}(l)]^2 \qquad (15)$$

where $\bar{d_c}(l)$ is the complement of the desired $d_c(l)$ ($\bar{d_c}(l) = 1 - d_c(l)$).

To compare the Kullback-Leibler error criterion with the LS measure, we need to express E_{KL} in terms of the distance between the MLP output and the desired signal. After some algebraic manipulation [11], E_{KL} can be rewrite as

$$E_{KL} = -\frac{1}{M}\sum_{l=1}^{M}\sum_{c=1}^{C}log[1 - |y_c(l) - d_c(l)|] \qquad (16)$$

For values of $|y_c(l) - d_c(l)| = |e_c(l)| << 1$ is approximately given by

$$E_{KL} = \frac{1}{M}\sum_{l=1}^{M}\sum_{c=1}^{C}|e_c(l)| \qquad (17)$$

Note that E_{KL} depends linearly on the difference between the actual output of the MLP and the desired output. It is this property of E_{KL} which makes it more suitable for the purpose of estimating posterior probabilities than the regular LS error measure (E_{LS}). As shown in (9), E_{LS} depends solely on the square of the error. Consequently, it overemphasizes the larger errors which makes it inappropriate for estimating the small posterior probabilities, as discussed earlier.

3. RESULTS

After traffic sign pre-processing stage a total number of 235 observation vectors is obtained, as indicated in 2.2. Each one is 63 samples length instead of 2700 (3*x*30*x*30) corresponding to size of the whole blob. It implies a reduction of 2637 inputs to the MLP, which reduces its computational cost.

The results obtained after training different MLPs of sizes $63/N/8$ (63 inputs, N hidden neurons, and 8 outputs) with the LS and Kullback-Leibler error criteria are shown in tables 1 and 2. As can be observed, the Kullback-Leibler error criterion provides better results than the LS one from two points of view. The first one is related to the MLP size necessary to obtain a certain number of errors. For the LS error criterion to achieve a minimum of 6 classification errors, a minimum of $N_{LS} = 33$ hidden neurons is necessary, whereas for the Kullback-Leibler one to achieve the same number of errors, only $N_{EN} = 15$ hidden neurons are

Table 1: Performance evaluation of different MLPs ($63/N/8$) trained with the LS error criterion

Hidden Neurons (N)	*Errors*	P_e
5	28	0.3590
7	22	0.2821
10	12	0.1538
15	12	0.1538
22	8	0.1026
33	6	0.0769
50	6	0.0769

Table 2: Performance evaluation of different MLPs ($63/N/8$) trained with the Kullback-Leibler error criterion

Hidden Neurons (N)	*Errors*	P_e
5	9	0.1154
7	8	0.1026
10	8	0.1026
15	6	0.0769
22	6	0.0769
33	5	0.0641
50	4	0.0513

needed. So the computational cost reduction obtained is about 55% ($|N_{EN} - N_{LS}|/N_{LS}x100$). The second point of view is related to the minimum P_e achieved. For the Kullback-Leibler error criterion, the minimum P_e obtained is $0,0513$ with a $63/50/8$ MLP. On the contrary, the system trained with the LS error criterion achieves a minimum $P_e = 0,0769$, which is greater than the obtained with the Kullback-Leibler one. The maximum MLP size ($63/50/8$) took as limit for the experiments is due to the theorem explained in [8][10], which relates the MLP size (MLP intelligence) needed to obtain a certain P_e and the number of observation vectors available.

It is important to note that the training parameters of the back-propagation algorithm (learning rate, momentum and its updating rates) are the same for both methods. This is the reason why they are not mentioned and explained along the paper.

4. CONCLUSIONS

A Traffic Sign Recognition system is presented in this paper which combines pre-processing and neural networks in the recognition core. Pre-processing allows to maintain the majority of the information of the input image, while reducing the dimension of the input vector that is applied to the neural network. This fact reduces the number of neural network weights and makes learning easier. The considered neural networks are Multilayer Perceptrons, trained to minimize two different error criteria. The first one is the well known Least Square error criterion. The second one is a modified Kullback-Leibler error measure.

Results obtained in this work show the robustness of the recognition stage against problems like lighting conditions, rotation, etc. These results provide low P_e for the best MLPs trained with the LS ($0,0769$) and Kullback-Leibler ($0,0513$) error criterions. So, an improvement of $0,0256$ ($33,2\%$) is obtained with the Kullback-Leibler one. Training with the Kullback-Leibler error criterion works better than with the LS one, due to the linear dependency of the Kullback-Leibler error criterion with the difference between the actual and the desired output. It permits to give the same importance to the estimations of the low and large posterior probabilities.

A reduction in the computational cost is achieved for the same error rate. The reduction got is about 55%, i.e., training a MLP with the Kullback-Leibler error criterion can reduce its processing time to the half of the corresponding to a MLP trained with the LS one.

REFERENCES

[1] Miura, J., et. al., "An active vision system for real-time traffic sign recognition", *Proc. IEEE Inteligent transportation systems*, 2000, pp. 52-57.

[2] Pérez, E. and Javidi, B., "Nonlinear distortion-tolerant filters for detection of road signs in background noise", *IEEE trans. on Vehicular Technology*, 2002, vol. 51, no. 3, pp. 567-576.

[3] Escalera, A. de la, et. al., "Visual sign information extraction and identification by deformable models for intelligent vehicles", *IEEE trans. on intelligent transportation systems*, 2004, vol. 5, no. 2, pp. 57-68.

[4] Hsu, S. H. and Huang, C. L., "Road sign detection and recognition using matching pursuit method", *Image and Vision Computing*, 2001, vol. 19, pp. 119-129.

[5] Escalera, A. de la, et. al., "Traffic sign recognition and analysis for intelligent vehicles", *Image and vision computing*, 2003, vol. 21, pp. 247-258.

[6] Fang, C., et. al, "Road Sign Detection and tracking", *IEEE trans. on vehicular technology*, 2003, vol. 52, no. 5, pp. 1329-1341.

[7] Fang, C. Y., et. al., "An automatic road sign recognition system based on a computational model of human recognition processing", *Computer Vision and Image Understanding*, 2004, vol. 96, pp. 237-268.

[8] S. Haykin, *Neural networks. A comprehensive foundation (second edition)*, Prentice-Hall Inc., London, 1999.

[9] F. Rosenblatt, *Principles of Neurodynamics*, Spartan books, New York, 1962.

[10] C.M. Bishop, *Neural networks for pattern recognition*, Oxford University Press Inc., 1995.

[11] A. El-Jaroudi, J. Makhoul "A new error criterion for posterior probability estimation with neural nets", *IJCNN International Joint Conference on Neural Networks*, 1990, vol. 3, pp. 185-192.

[12] S. Kullback, *Information Theory and Statistics*, Wiley, New York, 1959.