Committee C-Mantec: A Probabilistic Constructive Neural Network

Jose Luis Subirats, Rafael Marcos Luque-Baena, Daniel Urda, Francisco Ortega-Zamorano, Jose Manuel Jerez, and Leonardo Franco

Department of Computer Science, University of Málaga, Málaga, Spain
{jlsubirats,rmluque,durda,fortega,jja,lfranco}@lcc.uma.es

Abstract. C-Mantec is a recently introduced constructive algorithm that generates compact neural architectures with good generalization abilities. Nevertheless, it produces a discrete output value and this might be a drawback in certain situations. We propose in this work two approaches in order to obtain a continuous output network such as the output can be interpreted as the probability of a given pattern to belong to one of the output classes. The CC-Mantec approach utilizes a committee strategy and the results obtained both with the XOR Boolean function and with a set of benchmark functions shows the suitability of the approach, as an improvement over the standard C-Mantec algorithm is obtained in almost all cases.

Keywords: Committee networks, Supervised classification, Constructive neural networks.

1 Introduction

Neural computing techniques offer attractive alternatives over other classical techniques, particularly when data is noisy or in cases when no explicit knowledge is known. In practical applications, the most important criterion to evaluate the performance of trained Artificial Neural Networks (ANNs) is its ability to generalize knowledge. Although properly trained ANNs may offer very good results, they will inevitably overfit. Therefore, other techniques should be developed in order to improve ANNs generalization capabilities. A widespread approach involves training several networks (varying topologies, initialization of synaptic weights, etc.) and then choosing the one that offers the greatest generalization capacity. Under this approach, the acquired knowledge by non-optimal networks is lost, whereas, in principle, this information should not be discarded. One way to avoid this, is the use of a committee of ANNs, and it has been shown that by combining several ANNs the generalization can be improved [6]. The different approaches that exists for applying Committee machines can be classified into two broad categories [4]: Static and dynamic structures.

The generalization error generated by several ANNs that form a committee are not necessarily related. In this sense, when a committee based on different networks is created, the generalization error of a single ANN can be corrected by the remaining networks.

A very important issue related to the application of ANNs is the selection of a proper neural architecture for each network in the committee [1,3]. Despite the existence of several proposals to solve or alleviate this problem [4], there is no general agreement on the strategy to follow in order to select an optimal neural network architecture. Constructive algorithms have been proposed in recent years [5,8] with the aim of dynamically estimating the neural network topology. In general, constructive methods start with a small network, adding new units as needed until a stopping criteria is met. In [7], the Competitive MAjority Network Trained by Error Correction (C-Mantec) algorithm was introduced, with the novelty in comparison to existing approaches that C-Mantec incorporates competition between neurons and thus all neurons can learn at any stage of the procedure. Based on this previous algorithm, a new Committee C-Mantec (CC-Mantec) method is proposed in this work, in order to obtain a probabilistic version of the algorithm.

The remainder of this paper is organized as follows: Section 2 provides a description of C-Mantec algorithm, Section 3 shows two novel approaches, the HC-Mantec based in a Hyperbolic tangent sigmoid transfer function, and the CC-Mantec based in a Committee of networks. Section 4 shows the experimental results using several prediction, and finally, Section 5 concludes the article.

2 The C-Mantec Algorithm

C-Mantec is a constructive neural network algorithm that creates architectures containing a single layer of hidden nodes with sign activation functions. For binary classification tasks, the constructed networks have a single output neuron computing the majority function of the responses of the N hidden nodes:

$$CMantec\left(\boldsymbol{\psi}\right) = sign\left(\sum_{n=1}^{N} sign\left(h_n\left(\boldsymbol{\psi}\right)\right)\right) \tag{1}$$

$$h_n\left(\boldsymbol{\psi}\right) = \sum_{i=1}^{M} w_{n,i}\psi_i + \theta_n \tag{2}$$

$$sign\left(x\right) = \begin{cases} 1 & x \geq 0 \\ -1 & in\, other\, case \end{cases} \tag{3}$$

where M is the number of inputs of the target function, w_i are the synaptic weights, θ is the neuron threshold, and ψ_i indicates the set of inputs. The learning procedure starts with an architecture comprising a single neuron in the hidden layer and continues by adding a neuron every time the present ones are not able to learn the whole set of training examples. The hidden layer neurons learn according to the thermal perceptron learning rule proposed by Frean [2]. The thermal perceptron can be seen as a modification of the standard perceptron rule that incorporates a modulation factor, which forces the neurons to learn only target examples close to the already learned ones.

The network generated by the algorithm has an output neuron that computes the majority function of the activation of the neurons belonging to the single hidden layer. If the target of a given example is not matched by the network output, this implies that more than half of the neurons in the hidden layer classify incorrectly the current input. In these cases, the algorithm, in the training phase selects one of the 'wrong' neurons in a competitive process in order to retrain it. For a deeper analysis of the C-Mantec algorithm, see the original paper[7].

3 Probabilistic C-Mantec Approaches

In this section, two different versions of a C-Mantec algorithm with continuous output are proposed, namely: Hyperbolic tangent sigmoid C-Mantec (HC-Mantec) and Committee C-Mantec (CC-Mantec).

3.1 HC-Mantec

HC-Mantec is a simple continuous version of C-Mantec where a Hyperbolic tangent sigmoid transfer function is used in all the neurons of the hidden layer:

$$HCMantec\left(\boldsymbol{\psi}\right) = \frac{\sum_{n=1}^{N} tansig\left(h_n\left(\boldsymbol{\psi}\right)\right)}{N} \tag{4}$$

$$tansig\left(x\right) = \frac{2}{\left(1 + exp\left(-2x\right)\right)} - 1 \tag{5}$$

The output of the network is normalized by a factor equal to the the number of neurons of the hidden layer, N, so it belongs to the interval $[-1, 1]$.

3.2 CC-Mantec

CC-Mantec is a constructive neural network algorithm that uses the power of committee methods with the advantage of dynamically estimating each network architecture. C-Mantec can be seen as a neurons committee that finds one approximated solution to the problem, and in this sense, different executions of the algorithm would give different committee for solving the problem. Let K be the number of total neurons generated by all the single C-Mantec networks which compose our CC-Mantec approach. CC-Mantec use all K generated neurons to create a new single committee of neurons with no need to retrain this new model, such that, no relevant information is missed and, on average, the generated hyperplanes are quite close to the optimal solution (see Equation 6). Figure 1 shows the CC-Mantec network topology as a result of combining two single C-Mantec networks.

$$CCMantec\left(\boldsymbol{\psi}\right) = \frac{\sum_{n=1}^{K} sign\left(h_n\left(\boldsymbol{\psi}\right)\right)}{K} \tag{6}$$

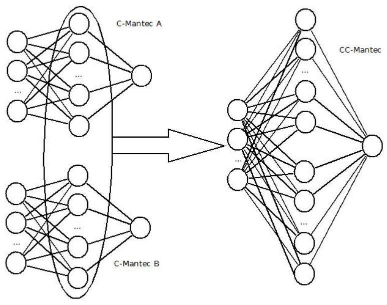


Fig. 1. The CC-Mantec topology obtained from combining two independent C-Mantec networks

3.3 Multiclass Classification

A $K - Class$ pattern recognition task can be implemented by a system formed by M CC-Mantec and an additional decision module. The M CC-Mantec are trained separately, and a decision module is used to select the final classification result based on the outputs of the M neural networks. The value of M and the training methodology depends on the modeling scheme used. In this work, three different approaches (One-Against-All, One-Against-One, and P-Against-Q) are applied.

One-Against-All scheme utilizes $M = K$ CC-Mantec, where K is the number of output classes of the original problem. Each CC-Mantec is trained with the same training dataset but with different objective values. On each of the K networks, one of the $K - classes$ is assigned the target value 1 while the rest of classes is assigned the value 0. The decision module computes the belonging probability for class i:

$$CCMantec_{OAA}(\mathbf{x}, i) = \frac{CCMantec_i\,(\mathbf{x})}{\sum_{j=1}^{M} CCMantec_j\,(\mathbf{x})} \tag{7}$$

One-Against-One scheme transforms a $K - Class$ pattern classification problem into $M = K(K-1)/2$ two-classes sub-problems. Each CC-Mantec solves a classification problem of an individual class against another and is trained only with a subset of the dataset where these two classes are active. A simple voting scheme can be used for the decision module, that computes the belonging probability for each class based on the continuous outputs from the M CC-Mantec networks.

$$CCMantec_{OAO}(\mathbf{x}, i) = \frac{\sum_{j=1, j\neq i}^{M} CCMantec_{i,j}\,(\mathbf{x})}{\sum_{i=1}^{M-1}\sum_{j=i+1}^{M} CCMantec_{i,j}\,(\mathbf{x})} \tag{8}$$

In the P-Against-Q classification scheme, the original classes are grouped in M different two class problems, in a way that from the output of these M groups is possible to infer the output class. The implementation can be considered as

M binary codes of length K, where each code has P bits equal to one and $Q = M - P$ bits equal to zero. One type of P-against-Q encoding consists in using the shorter code that specify all classes, $M = \log_2 K$ bits. This dense encoding is efficient in terms of the resulting size of the architecture but not in terms of the generalization obtained, as some redundancy on the encoding is usually beneficial. An Euclidean distance scheme is implemented by the decision module. The M CC-Mantec generate an output vector $\mathbf{v}$, and the class with code nearest to $\mathbf{v}$ will be the chosen output.

$$CCMantec_{PAQ}(\mathbf{x}, i) = \frac{Distance(\mathbf{v}, Code_i)}{\sum_{i=1}^{M} Distance(\mathbf{v}, Code_i)} \tag{9}$$

$$v_i = CCMantec_i\,(\mathbf{x}) \tag{10}$$

4 Experimental Results

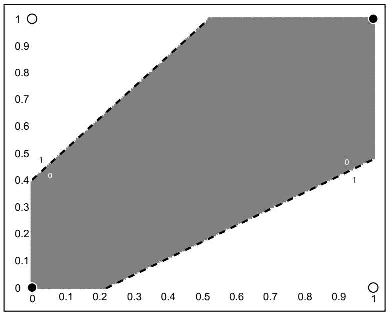
4.1 Detailed Analysis on the XOR Function

We have carried out a detailed analysis about the functioning of C-Mantec, HC-Mantec and CC-Mantec on the clasical XOR problem. Figure 2 represents some solutions obtained with the different approaches. The top two inset figures (Figures 2a and 2b), show two possible C-Mantec solutions, as different solutions are proposed depending on the order in which the training patterns are presented. These figures show that the 'zero' and 'one' classes are not balanced, since the position of the separating cutting planes are not optimal. In addition, the binary nature of the C-Mantec method gives no information about how close the points are to these planes. HC-Mantec (cf. 5) provides some improvement on the classification probabilities. Figures 2c and 2d show the belonging probability of each class in two possible solutions of the XOR problem. In this case, the method provides some information about how close is each point to the plane, but classes are still unbalanced. Figure 2e shows the result of the CC-Mantec approach. The figure was obtained using 2000 C-Mantec networks, and each represented point indicates the belonging probability to each class. For a better visualization purpose, the results were discretized in 10 regions and shown in Figure 2f, where it can be shown that the CC-Mantec output is very close to the optimal solution for the XOR problem.

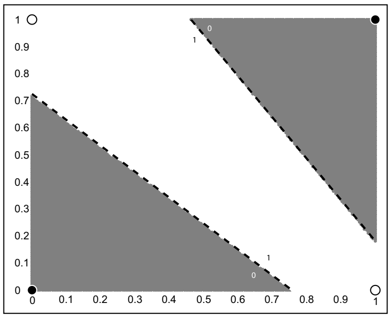
4.2 Tests on Benchmark Datasets

Seven benchmark data sets were used to analyze the performance of the introduced CC-Mantec algorithm, in comparison to the standard C-Mantec version.

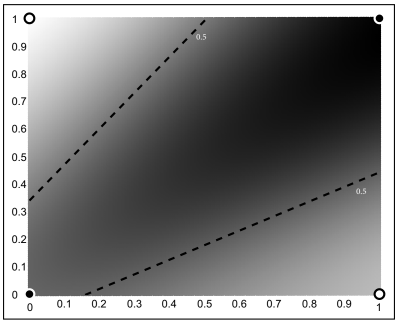
Table 1 and 2 shows the generalization results for the CC-Mantec and C-Mantec algorithms. For every data set, three multiclass approaches have been launched (OAA, OAO and PAQ), and the prediction accuracy was calculated as the average from 50 independent runs using a 80/20 training/generalization splitting sets with the standard C-Mantec parameter setting ($I_{max} = 10000, g_{fac} = 0.05$ & $\phi = 2$).



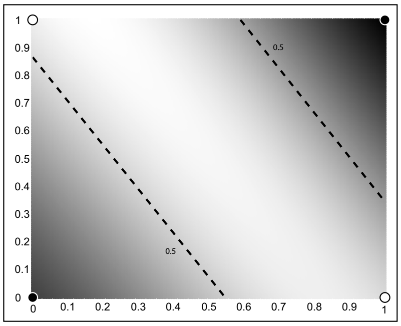
(a) A possible C-Mantec solution to the XOR problem.



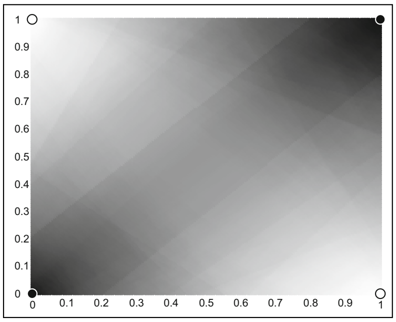
(b) Another C-Mantec possible XOR solution.



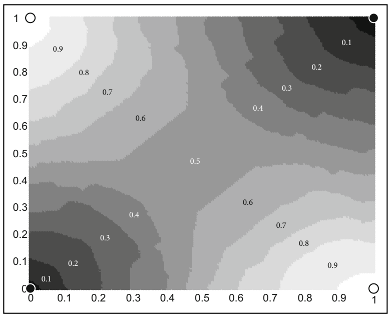
(c) HC-Mantec possible solution.



(d) Another HC-Mantec possible solution.



(e) CC-Mantec solution.



(f) CC-Mantec result discretized in 10 levels for better visualization.

Fig. 2. Solutions to the XOR problem obtained by the C-Mantec, HC-Mantec and CC-Mantec algorithms

Table 1. Results obtained by our CC-Mantec approach over several benchmark datasets. The generalization rate is shown using the mean and standard deviation.

	CC-Mantec		
	OAA	OAO	PAQ
balance-scale	91.392 ± 1.70	90.896 ± 2.57	91.488 ± 1.94
glass	**68.698 ± 6.17**	58.512 ± 8.39	59.163 ± 7.60
iris	**96.067 ± 3.51**	96.800 ± 2.98	93.733 ± 4.14
soybean	**92.486 ± 2.20**	90.397 ± 2.72	88.307 ± 3.25
vowel	**88.283 ± 2.58**	88.141 ± 2.58	56.626 ± 3.52
heart-statlog	**83.556 ± 4.10**	80.444 ± 5.30	83.407 ± 3.95
ionosphere	**88.732 ± 3.40**	83.408 ± 3.25	85.606 ± 3.31

Table 2. Results obtained by our C-Mantec approach over several benchmark datasets

	C-Mantec		
	OAA	OAO	PAQ
balance-scale	90.576 ± 2.20	89.872 ± 3.00	**92.432 ± 2.21**
glass	65.023 ± 7.51	66.930 ± 7.95	60.605 ± 6.87
iris	95.533 ± 3.10	95.933 ± 3.35	94.333 ± 3.61
soybean	91.037 ± 2.47	90.600 ± 3.20	81.861 ± 2.35
vowel	79.141 ± 4.19	87.182 ± 2.68	71.980 ± 3.63
heart-statlog	79.593 ± 4.62	79.778 ± 4.81	75.444 ± 5.50
ionosphere	87.465 ± 3.76	88.028 ± 3.51	86.620 ± 4.74

5 Conclusion

We propose in this work two possible extensions to the C-Mantec algorithm in order to obtain a continuous output value that approximates the probability of a given pattern to belong to one of many possible classes. The first proposal named HC-Mantec uses a sigmoidal activation function in the output layer to obtain a continuous response, but does not really work differently to the standard C-Mantec algorithm. The second proposal, named CC-Mantec, tries to take advantange of the potential of committee networks, and creates a network that combines several independent trained C-Mantec neurons, whose outputs are combined to obtain a continuous value in the range [-1, 1]. A detailed analysis of the performance of the new algorithms on the classic XOR problem is presented, showing that the behaviour of the CC-Mantec is very close to the optimal solution expected, as the output value can be interpreted as the probability of a given input pattern to belong to one of the output classes.

Using a set of 7 multiclass benchmark functions from the UCI repository an evaluation of the generalization ability of the CC-Mantec is carried out and compared to the standard C-Mantec implementation (results from the

HC-Mantec were not shown as they were almost indistinguishable with those from C-Mantec). The results show that in 6 out of the 7 data sets the CC-Mantec lead to a clear improvement in predictive accuracy (3.8 % of average improvement), suggesting the suitability of the developed approach. Some preliminary tests (not shown) done with other alternative classification algorithms (J48, SVM, Naive Bayes & MLP) confirms also that CC-Mantec performs better or at the level of bagging versions of these alternative algorithms, but these results will be the subject of further studies.

Acknowledgements. The authors acknowledge support through grants TIN2010-16556 from MICINN-SPAIN and, P08-TIC-04026 and P10-TIC-5770 (Junta de Andalucía), all of which include FEDER funds.

References

1. Baum, E.B., Haussler, D.: What size net gives valid generalization? Neural Comput. 1(1), 151–160 (1989)
2. Frean, M.: A thermal perceptron learning rule. Neural Comput. 4(6), 946–957 (1992)
3. Gómez, I., Franco, L., Jerez, J.M.: Neural network architecture selection: can function complexity help? Neural Process. Lett. 30(2), 71–87 (2009)
4. Haykin, S.: Neural Networks: A Comprehensive Foundation. Macmillan, New York (1994)
5. do Carmo Nicoletti, M., Bertini Jr., J.R.: An empirical evaluation of constructive neural network algorithms in classification tasks. Int. J. Innov. Comput. Appl. 1(1), 2–13 (2007)
6. Perrone, M.P., Cooper, L.N.: When networks disagree: Ensemble methods for hybrid neural networks, pp. 126–142. Chapman and Hall (1993)
7. Subirats, J.L., Franco, L., Jerez, J.M.: C-mantec: A novel constructive neural network algorithm incorporating competition between neurons. Neural Netw. 26, 130–140 (2012)
8. Subirats, J.L., Jerez, J.M., Franco, L.: A new decomposition algorithm for threshold synthesis and generalization of boolean functions. IEEE Trans. on Circuits and Systems 55-I(10), 3188–3196 (2008)