

Enhancing Decision-Based Neural Networks Through Local Competition

Gustavo Camps-Valls ^{a,1}, Luis Gómez-Chova ^a, Joan Vila-Francés ^a,
José D. Martín-Guerrero ^a, Antonio J. Serrano López ^a,
Emilio Soria-Olivas ^a

^a*Dept. Enginyeria Electrònica, Universitat de València, Spain.*

Abstract

In this paper the Decision-Based Neural Network (DBNN) learning algorithm is modified to stimulate local competition. Performance is assessed in ten UCI databases, resulting in improved results at the expense of a relatively low increase of the computational burden.

Key words: Decision-based neural network; hierarchical network structure; competitive credit-assignment scheme; local competition; UCI database.

Classification codes: neural networks, signal analysis.

1 Introduction

Credit-assignment criteria is the fundamental guiding principle for a great variety of classification algorithms available in the literature (e.g. neural networks, decision trees, support vector machines [1]). A particularly interesting decision-driven algorithm for pattern recognition is the decision-based neural network (DBNN), which usually provides very fast and satisfactory learning performance, along with an easily scalable network's structure. However, different strategies are needed when dealing with highly overlapping distributions and/or issues on false acceptance/rejection, e.g. introduction of non-linear discriminant functions, fuzzy-decision neural networks, or modular networks (see [2,3] for full details).

¹ Correspondence address: Prof. Gustavo Camps-Valls. Escola Tècnica Superior d'Enginyeria (ETSE). Dept. Enginyeria Electrònica. Grup de Processament Digital de Senyals. C/ Dr. Moliner, 50. Burjassot (València). Spain. Tel.: +34 96 3160197; Fax: +34 96 3160466. E-mail address: gustavo.camps@uv.es, <http://www.uv.es/~gcamps>.

In the DBNN framework, multi-classification problems are tackled by means of task division, and thus dedicated models (made up of a set of local subnets) to each class are implemented [2]. Then, learning is guided through a reinforced (antireinforced) learning applied to the subnet corresponding to the winner (loser) class, thus giving a prize (penalization) in the way of a negative (positive) gradient on the error surface, $\Delta \mathbf{w} = \pm \eta \nabla \phi(\mathbf{x}, \mathbf{w}) = \pm \eta \left[\frac{\partial \phi}{\partial \mathbf{w}_1}, \dots, \frac{\partial \phi}{\partial \mathbf{w}_P} \right]^\top$, where P is the total number of parameters, η is a positive learning rate, and ϕ represents the discriminant decision function.

The formal updating rule of the hierarchical structure of a DBNN is as follows. Given a labelled training data set $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(M)}\}$, each one corresponding to an output class $\{\mathcal{Y}_i, i = 1, \dots, L\}$. Each class is modelled by a subnet with discriminant (or transfer) functions $\phi(\mathbf{x}, \mathbf{w}_i)$, $i = 1, \dots, L$, which is formed by local basis functions (BFs) $\psi_l(\mathbf{x}, \mathbf{w}_{k_l})$, where the integer $k_l \in 1, \dots, K_l$, and K_l denotes the number of subnodes in the l th subnet [see Figure 1]. If the m th training pattern $\mathbf{x}^{(m)}$ belongs to class \mathcal{Y}_i , and $\phi(\mathbf{x}^{(m)}, \mathbf{w}_j^{(m)}) > \phi(\mathbf{x}^{(m)}, \mathbf{w}_l^{(m)})$, $\forall l \neq j$, the winning class for the pattern m is the j th class (or subnet). Therefore, two possibilities arise, which are summarized in the following scheme:

- (1) When $j = i$, then the pattern $\mathbf{x}^{(m)}$ is already correctly classified and thus, no update is necessary.
- (2) When $j \neq i$, then the pattern $\mathbf{x}^{(m)}$ is still misclassified and thus, the following two updates are performed:
 - (a) *Reinforced learning*:

$$\mathbf{w}_i^{(m+1)} = \mathbf{w}_i^{(m)} + \eta \nabla \phi(\mathbf{x}, \mathbf{w}_i) \quad (1)$$

- (b) *Antireinforced learning*:

$$\mathbf{w}_j^{(m+1)} = \mathbf{w}_j^{(m)} - \eta \nabla \phi(\mathbf{x}, \mathbf{w}_j) \quad (2)$$

Note that, for all $k \neq i$ and $k \neq j$, $\mathbf{w}_k^{(m+1)} = \mathbf{w}_k^{(m)}$, i.e. only the weights of the winner subnet and the correct class subnet are updated (the rest remain unchanged), which increases the speed of the learning process but can make the network to fall in local minima easily.

2 Subcluster Hierarchical DBNN

In order to further localize the training credit-assignment, the so-called *Subcluster Hierarchical DBNN* (SCH-DBNN) is commonly used [3] [see Fig. 1]. For the subcluster hierarchical structure, the concepts of local winner and global winner are introduced. The local winner s_l is the winner among the subnodes within the same l th subnet, $s_l = \text{Arg}\{\max_{k_l} \Theta_l(\mathbf{x}, \mathbf{w}_{k_l})\} \in \mathbb{Z}^+$. The global winner is the winner among all the subnets. The j th subnet will be labelled as the global winner if its local winner wins

over all the other local winners, $\Theta_j(\mathbf{x}, \mathbf{w}_{s_j}) > \Theta_l(\mathbf{x}, \mathbf{w}_{s_l}) \forall j \neq l$. This modification obviously involves substituting the discriminant function of the subnets by the local winners in equations (1) and (2), as follows: $\phi(\mathbf{x}, \mathbf{w}_i) \leftrightarrow \Theta_i(\mathbf{x}, \mathbf{w}_{s_i})$, and $\phi(\mathbf{x}, \mathbf{w}_j) \leftrightarrow \Theta_j(\mathbf{x}, \mathbf{w}_{s_j})$. Therefore, when \mathbf{x} is misclassified, antireinforced learning is applied to the local winner in the global winner subnet, and the reinforced learning is applied to the local winner in the correct class subnet.

3 Enhanced SCH-DBNN

Despite that the previous approaches deal with global and local credit-assignment issues efficiently, they are prone to some limitations because confidence/penalization relies only on inter-class competition, i.e. no in-class BFs responses are taken into account but the one from the local winner subnode. This limitation could lead to biased trainings if parameter η is not tuned correctly and, for many iterations, a node in a subnet becomes the local winner. This could limit the potential of the neural network and, in turn, increase the training time and efforts. In order to alleviate this problem, we introduce a local competition parameter that trades-off the inter-class against the in-class competition of BFs and, thus, assigns different penalization of BFs in the same subnet. The new algorithm is called Enhanced Subcluster Hierarchical DBNN (ESCH-DBNN) and reduces to the following two basic steps: (1) update the weights in the network only when an error is committed, and (2) the winner subnode s_l of the subnet l will be governed by the usual reinforced/antireinforced updates using the η learning rate. Additionally, the updating rules will also penalize errors of all loser subnodes by using a *leaky* learning rate, γ .

Notationally, let us now assume that the pattern $\mathbf{x}^{(m)}$ should belong to class i , but the j th subnet is selected as the global winner. In this general situation, the governing algorithm is as follows:

- (1) When $j = i$, then the pattern $\mathbf{x}^{(m)}$ is already correctly classified and thus, no update is necessary.
- (2) When $j \neq i$, then the pattern $\mathbf{x}^{(m)}$ is still misclassified and thus, the following two updates are performed:
 - (a) *Enhanced reinforced learning:*

$$\mathbf{w}_{s_i}^{(m+1)} = \mathbf{w}_{s_i}^{(m)} + \eta \nabla \Theta_i(\mathbf{x}, \mathbf{w}_{s_i}) \quad (3)$$

$$\mathbf{w}_{i \neq s_i}^{(m+1)} = \mathbf{w}_{i \neq s_i}^{(m)} + \gamma \nabla \Theta_i(\mathbf{x}, \mathbf{w}_{i \neq s_i}) \quad (4)$$

- (b) *Enhanced antireinforced learning:*

$$\mathbf{w}_{s_j}^{(m+1)} = \mathbf{w}_{s_j}^{(m)} - \eta \nabla \Theta_j(\mathbf{x}, \mathbf{w}_{s_j}) \quad (5)$$

$$\mathbf{w}_{j \neq s_j}^{(m+1)} = \mathbf{w}_{j \neq s_j}^{(m)} - \gamma \nabla \Theta_j(\mathbf{x}, \mathbf{w}_{j \neq s_j}) \quad (6)$$

Note that intuitively, the algorithm motivates nodes to become local winners both in winner (Eq. (3)) and loser sub-clusters (Eq. (4)). Additionally, the antireinforced learning is not only provided to the local winner in the correct class subnet (Eq. (5)), but also to their competing local nodes (Eq. (6)). In practice, good results are obtained using $\gamma < \eta$, which suggests that the leaky term γ corrects, rather than guides, the “extra” credit assigned to loser nodes throughout the network when using the η parameter solely in the standard SCH-DBNNs. Finally, note that the classical SCH-DBNN is a particular case of the proposed method when $\gamma = 0$.

4 Results

Ten databases from the UCI Machine Learning repository were selected to test the capabilities of our proposal [4]. The selected databases contain different numbers of features, input dimensions, and classes in order to analyze all possible situations of a given algorithm. In all cases, we used linear basis functions, which has demonstrated a good trade-off between generalization capabilities and simplicity elsewhere [3]. We varied the number of nodes in a subcluster (< 20 to avoid overfitting), the range of random weight initialization, and the learning rate (η between 0.01 and 3) in order to determine the best topology. In the case of the Enhanced SCH-DBNN, we additionally varied the leaky γ parameter between 0.001 and 0.1. Selection of the best subset of free parameters was done through 8-fold cross-validation in the training data set. Data were first normalized to give zero mean and unit variance.

Table 1 shows details on the selected databases, the reported results in [4], the obtained overall accuracy, OA[%], using the Standard and Enhanced SCH-DBNNs, and the average CPU time [s.] used for training. From this table, we can extract some preliminary conclusions: (1) Our proposed modification improves the results obtained using the standard algorithm in all databases; (2) the differences are slightly greater in problems with higher number of classes, which suggests that inter-class competition has been successfully enhanced; and (3) the differences are not numerically significant as the number of features increases, which makes our method especially convenient for large scale problems. In addition, the results obtained by our proposal are at the expense of a moderate increase in the computational burden. This result was expected since more comparisons and thus more weight updates must be performed. It is worth noting that the CPU time increases almost linearly with the number of classes, which is due to the fact that each class is modelled by a set of dedicated subnets.

Table 2 shows a comparison of the most updated results reported in the literature for some databases [5]. We can see that in most cases, the proposed method outperforms some of the state-of-the-art results provided by neural networks and support vector machines (ionosphere and sonar datasets), or decision trees (diabetes, ionosphere, and glass datasets).

5 Conclusions

In this paper, a straightforward but powerful modification of the subcluster hierarchical DBNN has been proposed. The idea relies on the concepts of local and global competitions, and includes additional reinforcement/antireinforcement rules at a local scale. We assessed various algorithms' performance in ten UCI databases. In general, results of the standard algorithm were improved at the cost of a relatively low increase of the computational burden. These outcomes suggest that our method could be useful in other credit-assignment and multi-expert learning schemes. At present, we are working on the extension and applicability of our method to other basis functions, such as radial-based kernel functions. The interested reader can obtain more details and further experimental results in [6].

References

- [1] J. Friedman, T. Hastie, R. Tibshirani, The elements of statistical learning. Data Mining, Inference and Prediction, Springer-Verlag, Berlin, Heidelberg, 2001.
- [2] S. Y. Kung, Digital Neural Networks, Prentice Hall, Englewood Cliffs, NJ, USA, 1995.
- [3] S. Y. Kung, J. S. Taur, Decision-based neural networks with signal/image classification applications, *IEEE Transactions on Neural Networks* 6 (1) (Jan 1995) 170–181.
- [4] C. L. Blake, C. J. Merz, UCI repository of machine learning databases, <http://www.ics.uci.edu/~mllearn/MLRepository.html> (1998).
- [5] W. Duch, Datasets used for classification: comparison of results, <http://www.phys.uni.torun.pl/kmk/projects/datasets.html> (2002).
- [6] G. Camps-Valls, L. Gomez-Chova, J. Vila-Francés, J. D. Martín-Guerrero, A. J. Serrano-López, E. Soria-Olivas, Analysis of Enhanced Decision-Based Neural Networks. Applications in UCI databases, channel equalization, texture classification and pharmacokinetics, Tech. Rep. TR-DIE-GPDS-08/23/2005, Dept. Enginyeria Electrònica, Universitat de València, Spain, available at <http://www.uv.es/~gcamps> (Aug 2005).

List of Figures

- 1 *The subcluster Hierarchical DBNN is based on a set of dedicated subnets (indexed by $l = 1, \dots, L$) to model a class or category, each consisting of multiple subnodes or subclusters in a subnet (indexed by k_l). These subnodes or basis functions, $\psi_l(\mathbf{x}, \mathbf{w}_{k_l})$, can be linear ($\psi_l(\mathbf{x}, \mathbf{w}_{k_l}) = \mathbf{x}^\top \mathbf{w}_{k_l} + \rho_l$, where ρ_l represents the bias for the l th subnetwork), polynomial ($\psi_l(\mathbf{x}, \mathbf{w}_{k_l}) = (\mathbf{x}^\top \mathbf{w}_{k_l} + \rho_l)^d$, where $d \in \mathbb{Z}^+$ and represents the order), or radial basis functions ($\psi_l(\mathbf{x}, \mathbf{w}_{k_l}) = \exp(-\|\mathbf{x} - \mathbf{w}_{k_l}\|^2 / 2\sigma_l^2)$, $\sigma_l \in \mathbb{R}^+$).* 7

List of Tables

- 1 *Characteristics and validation results of the selected UCI databases. From left to right columns: database name, number of patterns ($\#P$), number of features ($\#F$), number of classes ($\#C$), best reported overall accuracy (OA[%]) in [4], best overall accuracy obtained using the Standard and Enhanced SCH-DBNNs, and the average CPU time [s.] for the standard DBNN and our proposal.* 8
- 2 *Benchmark with the most updated overall accuracies (OA[%]) (see [5]) provided by different methods in a subset of the databases selected.* 9

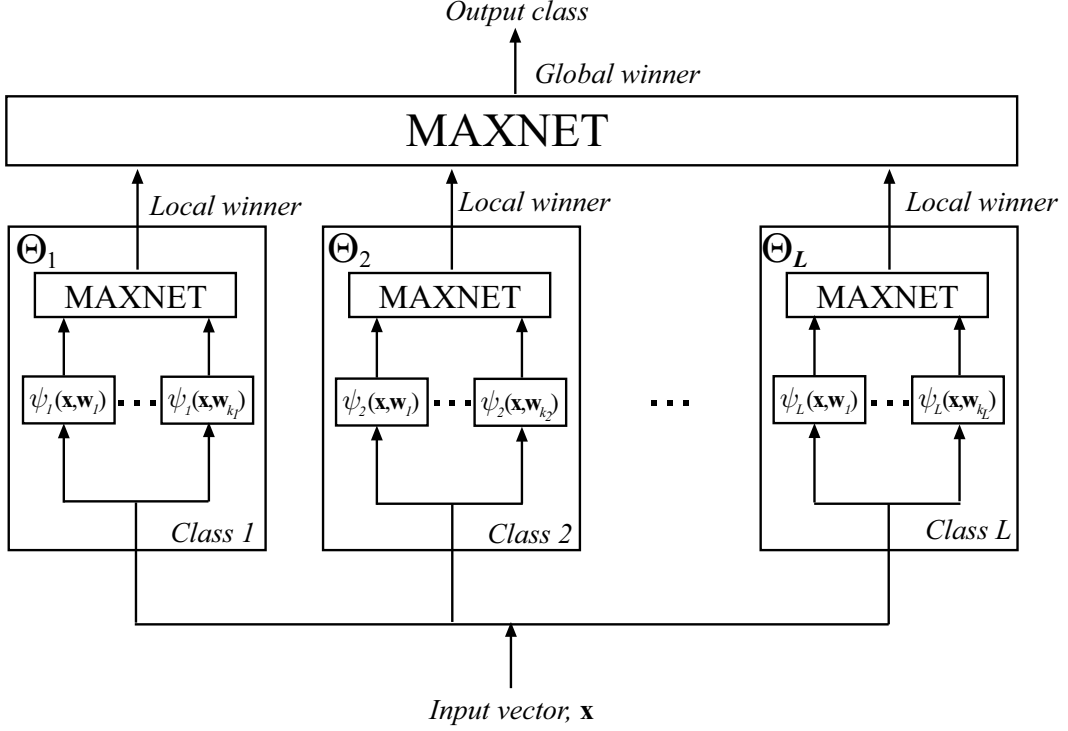


Fig. 1. The subcluster Hierarchical DBNN is based on a set of dedicated subnets (indexed by $l = 1, \dots, L$) to model a class or category, each consisting of multiple subnodes or subclusters in a subnet (indexed by k_l). These subnodes or basis functions, $\psi_l(\mathbf{x}, \mathbf{w}_{k_l})$, can be linear ($\psi_l(\mathbf{x}, \mathbf{w}_{k_l}) = \mathbf{x}^\top \mathbf{w}_{k_l} + \rho_l$, where ρ_l represents the bias for the l th subnetwork), polynomial ($\psi_l(\mathbf{x}, \mathbf{w}_{k_l}) = (\mathbf{x}^\top \mathbf{w}_{k_l} + \rho_l)^d$, where $d \in \mathbb{Z}^+$ and represents the order), or radial basis functions ($\psi_l(\mathbf{x}, \mathbf{w}_{k_l}) = \exp(-\|\mathbf{x} - \mathbf{w}_{k_l}\|^2 / 2\sigma_l^2)$, $\sigma_l \in \mathbb{R}^+$).

Table 1

Characteristics and validation results of the selected UCI databases. From left to right columns: database name, number of patterns ($\#P$), number of features ($\#F$), number of classes ($\#C$), best reported overall accuracy (OA[%]) in [4], best overall accuracy obtained using the Standard and Enhanced SCH-DBNNs, and the average CPU time [s.] for the standard DBNN and our proposal.

DATA BASE	#P	#F	#C	ACCURACY [%]			CPU TIME [s.]	
				REPORTED	SCH	ESCH	SCH	ESCH
				IN [4]	DBNN	DBNN	DBNN	DBNN
wdbc	569	30	2	-	97	98	34.10	40.40
glass	214	10	7	-	69	73	14.98	12.55
ionosphere	351	34	2	97	96	98	23.88	33.88
iris	150	4	3	-	95	97	1.80	2.25
liver-disorders	345	6	2	-	59	61	4.14	5.13
pima-diabetes	768	8	2	76	75	77	12.28	16.77
sonar	208	60	2	83	90	92	24.96	28.25
vehicle	846	18	4	-	68	74	60.92	71.15
vowel	990	10	11	56	71	81	108.91	150.54
wine	178	13	3	100	95	100	6.94	7.96

Table 2

Benchmark with the most updated overall accuracies ($OA[\%]$) (see [5]) provided by different methods in a subset of the databases selected.

pima-diabetes		ionosphere		sonar		glass	
Method	Acc.	Method	Acc.	Method	Acc.	Method	Acc.
SVM	77.6	3-NN + simplex	98.7	1-NN	97.1	Adaptive metric NN	75.2
ESCH-DBNN	77.1	ESCH-DBNN	98.3	TAP MFT Bayesian	92.3	ESCH-DBNN	73.3
Semi-Naive Bayes	76.0	MLP+BP	96.0	ESCH-DBNN	92.1	Discriminant Adaptive NN	72.9
C4.5	76.0	C4.5	94.9	SVM	90.4	kNN	72.0
Naive Bayes	74.5	SVM	93.2	MLP+BP	90.4	C4.5	68.2