Designing Partially-connected, Multilayer Perceptron Neural Nets through Information Gain

D. Rodríguez-Salas, P. Gómez-Gil and A. Olvera-López

Abstract—**An adequate number of hidden neurons and connection structure of a multi-layer perceptron network (MLP) are usually determined by experimentation. In this paper, we propose a scheme to define an appropriate structure and number of neurons of a partially connected MLP when used for classification. Rules for designing the network are based on a decision tree previously built using information gain. Our structure, called IG Net, is inspired by the Entropy Net [1], but contains fewer layers and connections than such network or than a fully-connected neural network and holds equivalent classification power. We tested the classification performance of our network using 10 databases from the UCI Machine Learning Repository. The performance obtained by IG Net using such databases showed to be statistically equivalent to the one obtained by an Entropy Net or by a fully-connected MLP, using fewer computational resources than the compared models.**

I. Introduction

MLP networks are widely used for classification due to their abilities to produce Input-Output Mappings (IOM) [2]. One of the main advantages of multi-layer perceptrons (MLPs) over other kinds of supervised classifiers is their generalization ability, that is, they are able to classify samples that are not part of the training set. However, in some cases the computational cost required to train MLPs is much higher than that required using other classifiers. A way to reduce this training cost is to build a network with a small but adequate number of hidden neurons and connections. Some effort has been done to find a way to define the right number of hidden neurons and strategies to eliminate connections among neurons [1,3-8] but still there is no easy way to do so [9]. In this paper we present a scheme for designing a partially connected neural network classifier, based on information gain, which is a metric used in the construction of decision trees. The proposed methodology is an improvement of the Entropy Net [1] where the idea of using entropy and decision trees to define network topologies was first proposed. Entropy nets are feed-forward networks with two hidden layers, known as the "partitioning layer" and the "ANDing layer." Such topology resembles the functions of a decision tree, which is previously built using a portion of the available data. The main advantage of Entropy nets is that they make classifiers showing the generalization capabilities of MLPs, but they also take advantage of the learning and classification speed of decision trees, using them for trimming the net structure. Notice that in a fully connected neural network classifier, each attribute (input node) is considered for the calculation of the output of each hidden node. Therefore, during the learning process, the relevance of each attribute has to be determined by all hidden neurons. On the other hand, the relevance of each attribute in a decision tree is calculated analyzing it just once, during the building of the tree. The construction of a decision tree allows finding discriminatory properties of attributes; the same idea may be considered a tool to determine meaningful connections in an MLP. The main difference between the IG Net and the Entropy net is in the number of hidden layers. Our proposal generates a network with one hidden layer while Entropy net uses two hidden layers, which results in fewer computational costs for learning and classification. The experiments shown here empirically demonstrated that IG Net is at least as good for classification as a fully connected MLP with a similar number of neurons, and as an Entropy Net.

This paper is organized as follows: section II describes some work related to this research, in particular the Entropy Net. In section III the scheme for building an IG Net is detailed; section IV shows the results obtained when using IG Net for some classification problems. Finally, section V presents some conclusions and suggests future work for this research.

II. Related Work

The study of methods for reducing the number of connections and neurons in an MLP has been of interest for several years. Elizondo and collaborators [10] defined three methods for the reduction of connections: the first one, an ontogenic method, reduces the number of connections during the learning process. The second, a non-ontogenic method, makes a data analysis before the learning process to achieve the reduction. The last method, a hybrid method, uses a combination of both models. Kang et al. [5] presented a strategy that removes weights that contribute the least to the network outputs. This is achieved by identifying whether an input attribute is coupled to another attribute, that is, if it is associated with another attribute by multiplication or it is uncoupled, which means it is associated by addition. This

Manuscript received March 1, 2013.

D. Rodriguez-Salas is with the Computer Science Department, Polytechnic University of Puebla, Mexico. During the development of this work, Mrs. Rodriguez was with Autonomous University of Puebla, México, dalia.rodriguez@live.com.mx.

P. Gómez-Gil is with the Computer Science Department, National Institute of Astrophysics, Optics and Electronics, Puebla México (*corresponding author*, ph:+52-222-266-3100; pgomez@acm.org.)

A. Olvera-López is with the Computer Science Department, Autonomous University of Puebla, México, aolvera@solarium.cs.buap.mx.

identification is made during the learning process. If at least one of the inputs of the network is uncoupled, then it is possible to model a partially connected network. A drawback of this model is that the network has to be trained as many times as attributes are in the set. Furthermore, an amplification value has to be defined for each attribute.

Sethi proposed a method to define a MLP partially connected network, called Entropy net, based on the building of a binary decision tree [1]. This tree is built using a portion of the training set prior to modeling the MLP. Entropy net has two hidden layers, the first called "partitioning layer" and the second called "ANDing layer." The output layer is called "ORing layer." These layers get their names by an analogy with the process of classifying an unknown input using a decision tree. When an unknown object is input to a decision tree, it will travel from the root toward one of the leaf nodes that determines the object class. To reach a leaf node, the object attributes must comply with all the conditions of at least one of the tree rules which were obtained during the learning process; this process implies AND operations. Moreover, if the corresponding object's attribute does not comply with one rule, it may comply with another one, which would imply at least one OR operation. The rules for building an Entropy net are:

1) The number of neurons in the first hidden layer equals the number of internal nodes of the decision tree. Each of these neurons implements one of the decision operations of the internal nodes. This layer is called "partitioning layer".
2) All leaf nodes have a corresponding neuron in the second hidden layer, the "ANDing layer".
3) The number of neurons in the output layer is equal to the number of classes. This layer is the "ORing layer".
4) Each neuron in the ANDing layer is connected to the neuron with the same class in the ORing layer.
5) The connections between the neurons of the partitioning layer and the neurons of the ANDing layer implement the tree's hierarchy.

To illustrate this process, Figure 1 shows the Entropy network associated to the XOR problem built with the rules previously described.

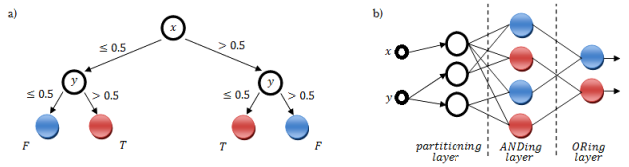

Fig. 1. (a) A decision tree for solving the XOR problem. (b) An Entropy generated using (a).

III. Design of the IG Net

The importance of an attribute in an MLP is represented with weight values assigned from the input neurons to each neuron in which the attribute is directly related (in the first hidden layer) or indirectly related (in the remaining layers). On the other hand, in a decision tree, the importance that provides an attribute to discriminate between classes is determined by the information gain of such attribute. Information gain of an attribute A, with respect to a training set T is defined as [11]:

$$Gain\,(T, A) = Entropy\,(T) - \sum_{v \in Values\,(A)} \frac{|T_v|}{|T|} Entropy\,(T_v) \quad (1)$$

$$Entropy\;\;(T) = \sum_{i=1}^{c} - p_i \log_2 p_i \quad (2)$$

where:

T_v : is the subset of T where A has value v.
p_i : is the proportion of samples belonging to class i.
c: Number of classes.

Equation (1) is commonly used for constructing a decision tree when defining the nodes at each level; the attribute with the maximum information gain is declared as the node for expanding.

In a decision tree, given a node p having m leaf nodes, there are m decision rules for assigning a class. Therefore, when classifying samples of the same class, the same path from the root is traversed until reaching the p node, that is, such samples are described with the same dimensional space. Using this idea, a topology can be proposed taking into account this relation between MLP and decision trees: the number of neurons can be related to the number of nodes in the tree and a path from the root to a leaf could be used to define the connections between hidden layers. IG net takes its name from the fact that rules for modeling it are based on the construction of a decision tree, which is built using information gain. In the following lines, the rules for defining the proposed IG net from a decision tree are described.

Let A be a decision tree built from a portion of the available data, representing the context to be classified, and let:

M_p : Be the number of parent nodes of at least one leaf node in A. In the case where the root node is parent of at least one leaf node, it is not taken into account,
p_i : the *i-th* parent node of a leaf, i=*1*.. M_p
m_i : the number of leaf nodes of p_i
k_{ij} : the class associated with the *j-th* leaf of p_i , $1 \le j \le m_i$

The next six rules define the topology of an IG net:

R1. The number of neurons in the input layer is the same

as the number of attributes involved in *A*.

R2. The number of neurons in the output layer is equal to the number of classes.

R3. The number of neurons in the hidden layer is equal to M_p. Each of these neurons, named g_i, have a corresponding tree node p_i.

R4. The inputs for each g_i are the attributes involved in the path from the root node to p_i.

R5. For the root node *r* of *A,* associated to the attribute x_j if *r* has n>0 leaf nodes, let K_{rj} be the class of the *i-th* leaf of *r,* for *i=1…n*. The neuron in the input layer corresponding to x_j is directly connected to the neurons in the output layer related to class K_{rj}.

R6. The output of each hidden neuron g_i feeds to neurons in the output layer with class equal to k_{ij} for *j=1…* m_i.

If one or more attributes are not involved in the rules of the decision tree, they will not be considered as inputs to the IG Net, because none of the subspaces defined by the tree will be determined by these attributes, as stated by rule **R1**. Given the fact that the rules in *A* generalize all the training samples, there is at least one leaf associated with each class on *T*, which explains **R2**.

It is important to remember that a MLP is able to represent an arbitrary IOM problem just using one hidden layer. This is proved through the universal approximation theorem [2,12]. For this reason, in the IG networks topology just one hidden layer is proposed as rule **R3** states**.**

The scheme for building an IG net tries to provide the network with a proper number of hidden neurons for learning the training set. To achieve this, a hidden neuron is assigned to each space defined by the decision tree, providing each neuron with the ability to generate hyper-planes in that space. This ability is obtained taking the attributes that describe such space as inputs of each neuron, which explains rule **R4**. When the root node is a parent of at least one leaf, a decision space is defined that depends on the attribute associated to the root, allowing the neuron associated with this node to have a single attribute as input. This task can be directly assigned to the neurons in the output layer corresponding to the classes involved in such space, thus obtaining rule **R5**. The neurons in the output layer are responsible for determining which neuron in the hidden layer better discriminates the belonging of an object to a specified class. To achieve this, the outputs of the hidden neurons feed the neurons in the output layer with the classes corresponding to their assigned decision space, which is expressed in the rule **R6**.

An example of the previously described rules is depicted in Figure 2, where each class is denoted in a different color. Figure 2(b) shows the IG Net derived from the decision tree in Figure 2(a). Notice that this IG net has only two hidden neurons, despite the existence of three parent nodes of leaves in the decision tree; this is due to rule R5. Figure 2(c) shows the Entropy network associated to the same tree. It can be noticed that the number of hidden neurons is the same as the number of nodes in the decision tree. The number of neurons in the hidden layer of in the IG networks directly depends on the tree from which it is built.

It must be pointed out that no pruning is applied during the construction of a decision tree for IG net definition. This is because the neurons in the hidden layer must consider all possible subspaces.

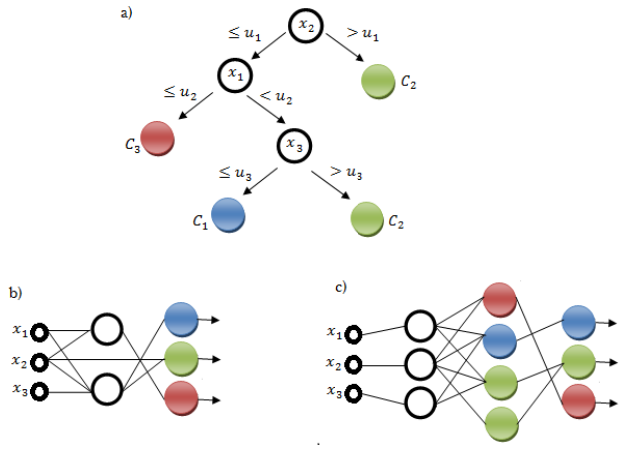

Fig. 2. (a) Decision tree for classification of 3 classes using 3 attributes. (b) IG network associated to tree in (a). (c) Entropy network associated to tree in (a).

IV. Results

Two experiments were designed to compare the performance of the proposed IG network with that obtained by a fully-connected MLP and by an Entropy network. The objective of the first experiment was to empirically verify if the classification performance of an IG network was at least equal or better than the other involved models. The second experiment aimed to verify whether or not the IG network reduces computational costs, which are measured in terms of training time, number of connections and number of hidden neurons. For both experiments, we used the Wilcoxon Signed-Rank test (WSR) [13] as a statistical significance test tool with a significance level of 0.05. Classifications were executed using 10 databases obtained from UCI Machine Learning Repository [14] (see Table 1). WSR test was chosen because it is suitable for the number of trials generated by the experiments reported here. All the attributes in the datasets were transformed to discrete values for constructing decision trees via the C4.5 algorithm [15]. Pruning was not applied for building of trees; trees were built using the whole training set, in order to avoid any bias in the classification performance. All MLP's were trained using a learning rate=0.3, momentum=0.2 and 1,000 epochs. All classifiers were built using toolboxes provided by

WEKA [16], as well as for building the trees, where the default input parameters were used.

The following network models were compared:

1. **IG net,** which corresponds to a partially-connected MLP designed using the rules proposed in this paper.
2. **Entropy net.** This is a MLP network built using the rules proposed in [1].
3. **MLP-IG**. A fully-connected MLP using one hidden layer, with the same number of hidden neurons as an IG network.
4. **MLP-Entropy.** A fully-connected MLP using the same number of layers and hidden neurons as an Entropy network.

A 10-fold, stratified cross-validation was applied for evaluating each classifier, using the 10 datasets listed in Table 1. Table 2 shows the average and standard deviation of the performances obtained for each classifier in each dataset using 10 folds. The best results by dataset are highlighted.

TABLE 1. DATASETS USED FOR THE EXPERIMENTS.

Dataset	Number of Objects	Number of Attributes	Number of Classes
Wine	178	13	3
Ionosphere	351	34	2
Iris	150	4	3
Pima Indians	768	8	2
Breast Cancer	699	9	2
Spambase	4,542	57	2
SPECT Heart	267	44	2
Connectionist	208	60	2
Statlog	231	19	7
Page Blocks	5,473	10	5

TABLE 2. AVERAGE PERFORMANCE USING 10-FOLD CROSS-VALIDATION

Dataset	IG net		Entropy Net		MLP - IG		MLP – Entropy	
	%	σ	%	σ	%	σ	%	σ
Wine	93.89	6.11	95.49	3.54	97.16	3.98	**97.75**	3.92
Ionosphere	**91.44**	4.27	90.31	8.5	89.75	4.89	90.60	4.68
Iris	**96.67**	4.71	95.33	4.50	95.33	8.92	95.33	8.92
Pima Indians	74.87	4.77	**77.08**	4.91	73.19	4.70	73.70	4.84
Breast Cancer	**95.84**	2.40	95.70	2.15	**95.84**	2.75	95.27	2.63
Spambase	**92.83**	1.52	**92.83**	1.85	90.04	4.57	92.18	1.16
SPECT Heart	80.17	6.33	**81.68**	5.28	78.65	1.7	77.14	6.61
Connectionist B.	79.31	6.43	77.45	8.36	**83.62**	7.60	79.31	10.34
Statlog	96.80	1.56	97.10	0.71	97.40	1.50	**97.45**	0.75
Page Blocks	96.55	0.72	**96.67**	0.73	96.53	0.75	96.64	0.77

A. First experiment: comparing classification performances

For the first experiment percentages of correctly classified objects, shown in Table 2, were used in a WSR test to compare the IG network with the rest of the models, determining if there existed a significant difference among them. This test was executed 10 times, one for each dataset. Only one significant difference was found between the IG network and the MLP-IG classification percentages, when using the "Spambase" dataset. For such execution, the best performance was obtained by the IG net and Entropy net. Therefore, for most cases, any of the four MLP network models obtained an average classification percentage with a non-significant difference with respect to the others.

B. Second experiment: comparing computational cost.

A WSR test was also used to find out if there was a significant difference in training time and in the number of connections among the four network classifiers. Table 3 shows the average training times of each network (t), measured in seconds, and the number of connections required for each classifier (c), both for each dataset. These execution times were obtained using a Pentium Dual-Core processor, 2.10 GHz, with 2GB of RAM and do not include the time for building the decision trees. Each entry in column (t) corresponds to the average of 10 training times, one for each fold. On the other hand, each entry in column (c) is a constant for each fold. This is because the number of connections depends upon the decision tree structure rather than the training data of the networks.

All results shown in Table 3 were used to execute the WSR test, obtaining the results shown in Table 4; the symbol "*" next to some models indicates a statistically significant difference of such model with respect to the paired one. In this experiment, only one WSR test is executed, using the 10 datasets. Notice that in this experiment the WSR test obtained a significant difference in the times and connections of the IG net with respect to MLP-IG and MLP Entropy, showing the IG network model required fewer connections and runtimes than the fully connected approaches.

Figure 3 compares the number of hidden neurons between IG net and Entropy net. There is no need to include MLP-IG network and MLP-Entropy network in this plot because the first has the same number of neurons as the IG Net and the second has the same number of neurons as Entropy Net. This figure, drawn in a logarithmic scale, shows that IG net has fewer hidden neurons than Entropy Net in all cases.

Given the fact that the classification percentage of the IG network model was statistically equivalent to the performance obtained by the other three models, the choice of a particular model depends on the computational cost required for each model. Taking in account the training time and the number of connections, it was found the IG network model is significantly better than the fully connected MLPs. This advantage is not obtained with the Entropy network because there was no significant difference among these values. However, the IG networks were significantly better in the number of hidden neurons, which considerably reduces the memory resources required.

TABLE 3. AVERAGE TRAINING TIME (T) (SECONDS) AND NUMBER OF CONNECTIONS (C) FOR EACH NETWORK.

Dataset	IG net		Entropy net		MLP - IG		MLP - Entropy	
	t	c	t	c	t	c	t	c
Iris	0.62	21	0.61	16	0.73	35	0.87	36
Wine	0.58	14	0.96	21	0.96	48	1.77	87
Connectionist	2.85	96	2.39	63	12.49	930	10.75	732
SPECT Heart	6.54	251	4.79	121	18.27	1058	17.52	932
Ionosphere	3.66	68	3.39	57	1.11	468	9.75	362
Breast Cancer	8.77	85	14.28	113	4.11	187	24.38	407
Pima Indians	24.89	288	41.02	355	33.53	460	114.88	1952
Statlog	88.75	281	142.31	445	218.1	1300	513.68	3017
Spambase	1636.23	3203	1530.60	2421	3499.69	12036	18252.56	36351
Page Blocks	333.17	499	475.92	133	560.4	1275	2156.26	5417

TABLE 4. SIGNIFICANT DIFFERENCES IN THE MODELS WITH RESPECT TO THE NUMBER OF CONNECTIONS AND TRAINING TIMES, OBTAINED WITH A WRS TEST. THE SYMBOL '*' SHOWS CASES WHERE A SIGNIFICANT DIFFERENCE WAS FOUND.

Connections		Time (seconds)	
IG net	Entropy net	IG net	Entropy net
IG net*	MLP-IG	IG net*	MLP-IG
IG net*	MLP-E	IG net*	MLP-Entropy

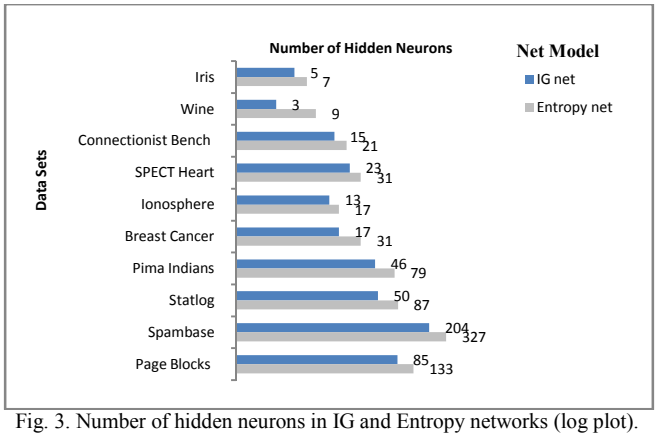

Fig. 3. Number of hidden neurons in IG and Entropy networks (log plot).

V. CONCLUSIONS

In this paper, we present a novel scheme to define a topology for a partially connected MLP when used for classification, called IG net, similar to the one defined by Sethi for the construction of an Entropy net [1]. When tested over 10 datasets, the performance in classification of our network was statistically equivalent to the performance of the fully-connected MLPs and to the performance of Entropy nets with a similar number of neurons.

Using an IG network, the training time and amount of memory used by a classifier is reduced because unnecessary connections are eliminated without a significant loss in the classification percentage Therefore, when using an IG net for a specific IOM problem, there is no need to decide the number of hidden layers and the number of hidden neurons, since these values are determined on the construction of the network's topology. This also eliminates training and testing networks with different numbers of neurons and hidden layers.

In this work, the IG net topology is based on discrimination rules from a decision tree. As future direction of our research we will explore some other approaches related to feature correlation and feature selection in order to discover relevant feature subsets to be considered as input/output in the layers of a MLP.

REFERENCES

[1] I.K. Sethi , "Entropy Nets: from decision trees to neural networks," *Proc. IEEE*, vol. 78, no. 3, pp. 1605-1613, 1990.

[2] S. Haykin, *Neural Networks and Learning Machines*, 3ª edition. Upper Saddle River, NJ: Pearson, 2009.

[3] D.E. Duckro, D.W. Quinn and S.J. Gardner, "Neural network pruning with Tuckey Kramer multiple comparison procedure," *Neural Computation*, vol. 14, no. 5, pp. 1149–1168, 2002.

[4] S.I. Sulaiman, T.K. Abdul Rahman and I. Musirin, "Optimizing one-hidden layer neural network design using evolutionary programming," *Proc. of the 5th International Colloquium on the Signal Processing & Its Applications (CSPA 2009)*, March 2009, pp.288-293, doi: 10.1109/CSPA.2009.5069236

[5] S. Kang and C. Isik. "Partially connected feed-forward neural networks structured by input types", *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 175-184, 2005.

[6] J.T Tsai, J.H Chou and T.K. Liu, "Tuning the structure and parameters of a neural network by using hybrid Taguchi-genetic algorithm," *IEEE Transactions on Neural Networks*, vol.17, no.1, pp.69-80, Jan. 2006, doi: 10.1109/TNN.2005.860885.

[7] P. Lauret, E. Fock and T. A. Mara, "A node pruning algorithm based on a Fourier amplitude sensitivity test method," *IEEE Transactions on Neural Networks*, vol.17, no.2, pp.273-293, March 2006, doi: 10.1109/TNN.2006.871707.

[8] C. Xiao, Z. Cai, Y. Wang and X. Liu, "Tuning of the structure and parameters of a neural network using a good points set evolutionary strategy," *Proc. of the 9th International Conference for Young Computer Scientists, 2008 (ICYCS 2008),* 18-21 Nov. 2008, pp.1749-1754, doi: 10.1109/ICYCS.2008.187

[9] D. Hunter, H. Yu, M. S. Pukish, J. Kolbusz and B. M. Wilamowski "Selection of proper neural network sizes and architectures: A Comparative Study", *IEEE Transactions on Industrial Informatics*, vol. 8, no. 2, pp. 228-240, 2012.

[10] D. Elizondo and E. Fiesler, "A survey of partially connected neural networks", *Int. J. Neural Syst.*, vol. 8, no. 5 y 6, pp. 535-558, 1997.

[11] T. M. Mitchell, *Machine learning*. McGraw Hill, 1997, ch. 3.

[12] G. Cybenko, "Approximation by superposition of a sigmoidal function", *Mathematics in Control, Signals and Systems*, vol. 2, pp. 303-314, 1989.

[13] J. Demšar, "Statistical comparisons of classifiers over multiple data sets", *The Journal of Machine Learning Research*, vol.7, pp.1-30, 2006.

[14] K. Bache and M. Lichman. *UCI Machine Learning Repository,* CA: University of California, School of Information and Computer Science. Available: http://archive.ics.uci.edu/ml.

[15] J.R. Quinlan, *C4.5: Programs for Machine Learning*, San Mateo, CA: Morgan Kaufmann Publishers, 1993.

[16] I.H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques.* Burlington, MA: Morgan Kaufmann Publishers, 2012, ch. 10.