



Alleviating the complexity of the Combinatorial Neural Model using a committee machine

H. A. do Prado^{1,2,3}, K. F. Machado^{1,4} & P. M. Engel¹

¹Federal University of Rio Grande do Sul, Brazil

²Brazilian Enterprise for Agricultural Research, Brazil

³Catholic University of Brasília, Brazil

⁴Justice Court of State of Rio Grande do Sul, Brazil

Abstract

Knowledge Discovery from Databases (KDD) can be seen as a set of computer-aided knowledge discovery techniques scaled up to very large databases. By this way, the old process of discovery has experienced amazing improvements by: (a) allowing well-known Machine Learning and Statistical algorithms run for larger data sets with good performance; and (b) making easier tasks like data gathering and cleansing, parameter and model selection, and so on. In this paper we take the Combinatorial Neural Model (CNM), proposed by Machado and Rocha ([6], [7], and [8]), and explore the adoption of a committee machine to cope with the complexity problem present in this model. In our proposal, a static committee machine is built on a certain number of CNMs. The committee machine takes the discrete outputs from the individual models, and proceeds by choosing the most accurate output using a voting process. Many works has been presented to alleviate the complexity problem in CNM and this one represents an alternative that can be combined with other solutions. The results from the experiments point out that, by means of a committee machine, one can reach a better trade-off between predictive accuracy and complexity. To evaluate the effectiveness of the technique, we apply it to some well-known data sets from the UCI repository [2]. Finally, it is suggested some extensions to improve the committee machine in order to obtain a better generalization performance, towards a universal approximator.



1 Introduction

Scaling up machine learning algorithms to run over very large databases and making easier tasks like data cleansing and parameter selection are two important approaches for Knowledge Discovery from Databases (KDD). In this sense, Combinatorial Neural Model (CNM) has received, recently ([1], [3], [10], and [11]), many improvements that have turned it suitable for KDD.

This paper presents an implementation of a committee machine based on CNM and reports some preliminary results that can be used to alleviate the complexity problem of CNM.

Next section presents a rough description of committee machines. Section 3 explains in more detail the CNM. Section 4 describes the method applied to build the committee machine and the files used to run it. The fifth section analyzes the results obtained and the last one draws some conclusions and states some paths to explore as future work.

2 Committee machines

The central idea underlying a committee machine is that a set of experts, in consent, are expected to make better predictions than each one separately. According to Haykin [5], committee machines are universal approximators and apply the *principle of divide and conquer* to face computationally complex problems. Many works has been reported (*e.g.*, [9] and [12]) that account for the applicability of committee machines to reach better generalization performance. Committee machines may be classified as static or dynamic structures. As a static structure, it takes the responses from many predictors, combining them into a final solution, not involving the input signal. In the dynamic committee machine the input signal is involved and actuates directly in the mechanism that integrates the individual experts. The particular kind of committee machine adopted in this work is the *ensemble averaging method*. In this method, a set of experts (models) is trained in parallel and their outputs are combined somehow to produce an overall output. This method presents two interesting advantages [5]: the first one is regarded to the training time. If a unique expert replaces the experts in the committee machine, the training time is expected to be higher than the sum of the times for all experts. The second advantage is that the risk of overfitting the data decreases when you take separate portions of the data set and combine them in a committee machine.

3 Combinatorial neural model

CNM is a hybrid architecture for intelligent systems that integrates symbolic and connectionist computational paradigms. It has some significant issues, such as the ability to build a neural network from background knowledge; incremental learning by examples, solving the plasticity-stability dilemma [4]; a way to cope with the diversity of knowledge; knowledge extraction of an

artificial neural network; and the ability to deal with uncertainty. CNM is able to recognise regularities from high-dimensional symbolic data, performing mappings from this input space to a lower dimensional output space.

CNM uses supervised learning and a feedforward topology with one input layer, one hidden layer - here called combinatorial - and one output layer (Figure 1). Each neuron in the input layer corresponds to a concept - a complete idea about an object of the domain, expressed by an object-attribute-value form, they represent the evidences of the domain application. In the combinatorial layer there are aggregator type neurons, each one connected to one or more neurons in the input layer by fuzzy AND arcs that represent logical concepts. The output layer contains one neuron for each possible class (also called hypothesis), linked to one or more neurons in the combinatorial layer by fuzzy OR arcs that also represent concepts. The synapses may be excitatory or inhibitory and they are characterised by a strength value (*weight*) between zero (not connected) to one (fully connected synapses), that can express the logical relations. For the sake of simplicity, we will work with the learning of crisp relations, thus with strength value of synapses equal to one, when the concept is present, and zero, when the concept is absent. However, this option does not affect the approach to fuzzy relations learning.

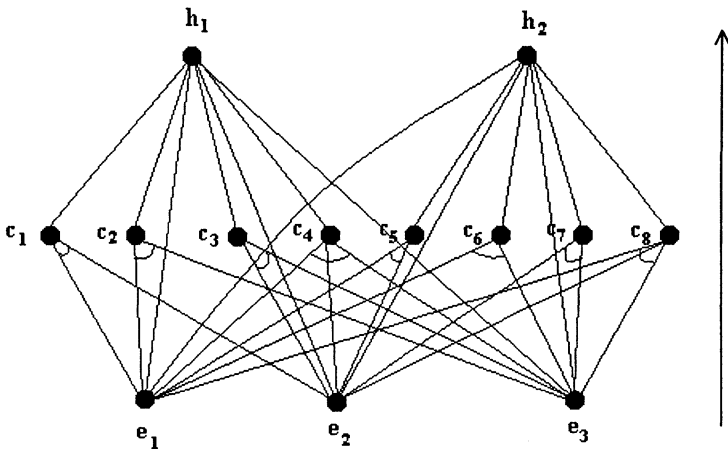


Figure 1. Complete version of CNM for 3 input evidences and 2 hypotheses [7]

The network is created completely empty, according to the following steps: (a) one neuron in the input layer for each evidence in the training set; (b) a neuron in the output layer for each class in the training set; and (c) for each neuron in the output layer, there is a set of hidden neurons in the combinatorial layer. The combinatorial layer comprises all possible combinations (length between two and nine) of the neurons in the input layer for each class in the output layer. There is no neuron in the combinatorial layer for length one connections. In this case, input neurons are connected directly to hypotheses.



The learning mechanism works in only one iteration, and it is described below:

PUNISHMENT_AND_REWARD_LEARNING_RULE

- **Set** to each arc of the network an accumulator with initial value of zero;
- **For** each example case from the training database, **do**:
 - **Propagate** the evidence beliefs from input nodes until the hypotheses layer;
 - **For** each arc reaching a hypothesis node, **do**:
 - If** the reached hypothesis node corresponds to the correct class of the case
Then backpropagate from this node until the input nodes, increasing the accumulator of each traversed arc by its evidential flow (Reward)
Else backpropagate from the hypothesis node until the input nodes, decreasing the accumulator of each traversed arc by its evidential flow (Punishment).

After training, the value of the accumulators associated to each arc arriving to the output layer will be between $[-T, T]$, where T is the number of all cases present in the training set.

The last step is the pruning of the network. This is performed by the following actions: (a) remove all arcs whose accumulator is lower than a threshold (specified by a specialist); (b) remove all neurons from the input and the combinatorial layers that became disconnected from all hypotheses in the output layer; and (c) make weights of the arcs arriving at the output layer equal to the value obtained by dividing the arc accumulators by the largest arc accumulator value in the network. After pruning, the network becomes operational for classification tasks.

4 Material and methods

We used three files from UCI repository [2]:

1. Breast-Cancer obtained from the University Medical Centre, Institute of Oncology, and Ljubljana, Yugoslavia. (Sources: Matjaz Zwitter and Milan Soklic);
2. Balance Scale Weight and Distance, generated to model experiments reported by [13]; and
3. Monks-2 (donated by Sebastian Thrun, School of Computer Science, Carnegie Mellon University).

For each data set we took, at random, 4 samples of the same size (or almost the same size, depending whether they have an integer division by 4). We called these samples A, B, C, and D. Then we built four models, M1, M2, M3, and M123, using, respectively, A, B, C, and A+B+C. Moreover, we create the committee machine MC with the combination of M1, M2, and M3. The data set D remained reserved to be used as an independent test bed. After the training, each model was pruned, keeping only arcs arriving to the output layer with positive accumulators. In other words, only rules with a minimum empirical

support were kept. During the construction of the models, each model size - taking into account the input, output, and the combinatorial layers - and training time were recorded. Also, in the test phase, the accuracy (percentage of right predictions) was measured presenting each model with the data set D. The whole process is illustrated in Figure 2.

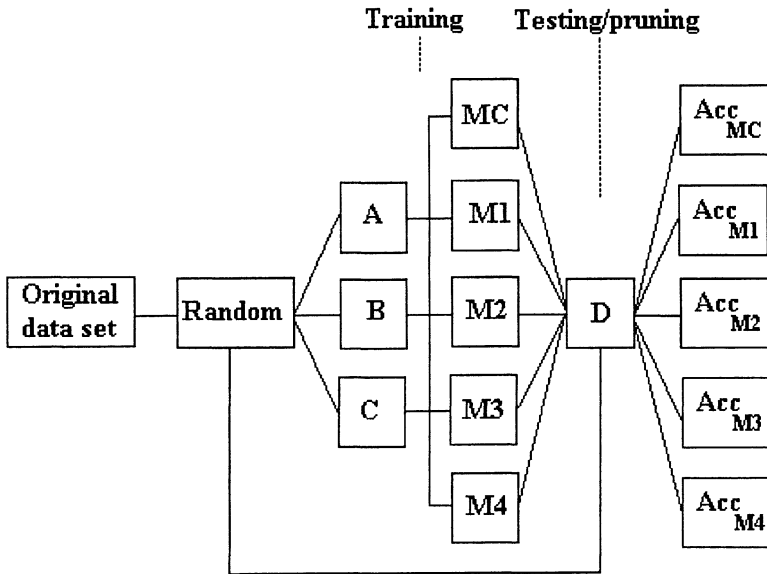


Figure 2. Model generation and testing process

5 Analysis of the results

This work presents some results towards the expansion of CNM to a universal approximator, using a committee machine. The results obtained with each model are summarized in Table 1.

It can be observed that using a committee machine, built from CNMs, as a classifier one can reach an accuracy comparable to the monolithic model M123. These results were obtained while reducing the complexity. In this sense, it was observed that each model, separately, in MC is much smaller than M123. Concerning to the training time, the results are more interesting. In this case, the sum of the training time of all components of MC is smaller than the training time of the whole M123. Both facts above point out that a committee machine represents an important alternative to cope with the temporal and spatial complexity of CNM.

On the other hand, although committee machines are considered universal approximators, this characteristic was not observed in the experiments. The adoption of a committee machine, as described in the previous section, did not improve the predictive accuracy of the final model.



Table 1. Summary of performance

		Models				
File	Item	M1	M2	M3	M123	MC
Breast Cancer	Size (nodes)	24,799	24,989	23,958	55,133	
	Training time (s)	407	401	380	2,918	
	Accuracy (%)	80.28	74.64	76.05	81.69	76.06
Balance Scale	Size (nodes)	1,014	1,025	997	1,822	
	Training time (s)	2	2	2	9	
	Accuracy (%)	81.4	76.92	85.89	86.53	85.25
Monks-2	Size (nodes)	2,744	2,781	2,782	3,833	
	Training time (s)	12	10	10	44	
	Accuracy (%)	87.70	90.98	91.80	95.08	94.26

Results shown in Table 1 point out the fact that, using a committee machine combined with a CNM, one can obtain a predictive accuracy only comparable to that obtained by the monolithic model (the model M123). It does not give the status of a universal approximator to MC. We believe MC can be improved to acquire this characteristic and present some suggestions in the next section.

6 Conclusions and future work

The committee machine has shown to be an interesting approach to face the complexity problem of CNM. Concerning to the expected improving of predictive accuracy of MC, the results were not satisfactory, since they just remained in the range of the component models. We are going to explore some directions in order to improve MC towards a universal approximator.

Our next step is to analyze the effects of taking the belief factors (given by the accumulators) of each rule in CNM, on deciding the class of an example in a MC. For instance, let us apply models M1, M2, and M3 to an example in the test data set. Suppose that the predicted classes, with the corresponding belief factors (BF), for each model are as following:

M1:	Class = X	(BF = 90%)
M2:	Class = Y	(BF = 60%)
M3:	Class = Y	(BF = 75%)

By the simple voting process, the result would be Class = Y. However, taking into account the belief factors, model M1 with BF = 90% would give Class = X (BF = 90%) as the final classification. Possibly, this result will be more accurate.

Another improvement we intend to do is the construction of the models in order to assign each one to a specific portion of the problem space. For instance, instead of building each model independently using random samples, we could



build the first model using a bigger sample, take the wrong predictions and use them to train the next model, and so on with the next model.

References

- [1] Beckenkamp, F. G., Feldens, M. A., Pree, W., Optimizations of the Combinatorial Neural Model. *Proc. of the Vth Brazilian Symposium on Neural Networks. (SBRN'98)*, Belo Horizonte, Brazil, 1998.
- [2] Blake, C. L. & Merz, C. J. UCI Repository of machine learning databases [<http://www.ics.uci.edu/~mllearn/MLRepository.html>]. Irvine, CA: University of California, Department of Information and Computer Science, 2000.
- [3] Feldens, M. A.: Engenharia da Descoberta de Conhecimento em Bases de Dados: Estudo e Aplicação na Área de Saúde Porto Alegre: CPGCC da UFRGS, Brasil, 1997. (M.Sc. Dissertation).
- [4] Freeman, J. A., Skapura, D. M.: Neural Networks, Algorithms, Applications, and Program Techniques. [S.l.]: Addison-Wesley, 1992. p.292-339.
- [5] Haykin, S., *Neural Networks: A Comprehensive Foundation*, Prentice Hall: Upper Saddle River, NJ, 842pp, 1999.
- [6] Machado, R. J., Rocha, A. F. Handling knowledge in high order neural networks: the combinatorial neural network. Rio de Janeiro, RJ: IBM Rio Scientific Center, 1989. (Technical Report CCR076).
- [7] Machado, R., Rocha, A., The combinatorial neural network: a connectionist model for knowledge based systems. *Uncertainty in Knowledge Bases*, eds. N: B. Bouchon, R. R. Yager, and L. A. Zadeh, Springer-Verlag: Berlin, pp.578-587, 1991.
- [8] Machado, R. J., Carneiro, W., Neves, P. A. Learning in the combinatorial neural model. *IEEE Transactions on Neural Networks*, **9**, pp.831-847, 1998.
- [9] Parmanto, Bambang, Munro, Paul W., Doyle, H.R., Improving Committee Diagnosis with Resampling Techniques. IN: D. Touretsky, M. Mozer, and M. Hasselmo (Eds.) *Advances in Neural Information Processing* **8**, MIT Press, 1996.
- [10] Prado, H. A., Frigeri, S. R., Engel, P. M., A Parsimonious Generation of Combinatorial Neural Model. *Proc. of the IV Congreso Argentino de Ciencias de la Computación (CACIC'98)*, Neuquén, Argentina, 1998.
- [11] Prado, H. A. do; Machado, K.F.; Frigeri, S. R.; Engel, P. M. Accuracy Tuning in Combinatorial Neural Model. *Proc. of Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Beijing, China, 1999.
- [12] Schapire, R. E., Freund, Y., and Bartlett, P., Boosting the margin: A new explanation for the effectiveness of voting methods. *Proc. of Fourteenth International Conference of Machine Learning*, Nashville, TN, USA, 1997.
- [13] Siegler, R. S. Three Aspects of Cognitive Development. *Cognitive Psychology*, **8**, 481-520.