An Algorithm for Extraction of Decision Trees from Artificial Neural Networks

Dr.M.Rajaiah[1], Dean Academics & HOD, Dept of CSE, Audisankara College of Engineering and Technology, Gudur.

Dr.N.Krishna Kumar[2], Associate Professor ,Dept of CSE, Audisankara College of Engineering and Technology, Gudur.

Mr. Akula Sujan Kumar[2], UG Scholar, Dept of CSE, Audisankara College of Engineering and Technology, Gudur.

Mr. Amruthala Anil Kumar[3], UG Scholar, Dept of CSE, Audisankara College of Engineering and Technology, Gudur

Mr. Kamineni Venkat Chowdary[4], UG Scholar, Dept of CSE, Audisankara College of Engineering and Technology, Gudur.

Ms. Addam Vennela[5], UG Scholar, Dept of CSE, Audisankara College of sEngineering and Technology, Gudur

ABSTRACT

Although artificial neural networks can represent a variety of complex systems with a high degree of accuracy, these connectionist models are difficult to interpret. This significantly limits the applicability of neural networks in practice, especially where a premium is placed on the comprehensibility or reliability of systems. A novel artificial neural-network decision tree algorithm (ANN-DT) is therefore proposed, which extracts binary decision trees from a trained neural network. The ANN-DT algorithm uses the neural network to generate outputs for samples interpolated from the training data set. In contrast to existing techniques, ANN-DT can extract rules from feedforward neural networks with continuous outputs. These rules are extracted from the neural network without making assumptions about the internal structure of the neural network or the features of the data. A novel attribute selection criterion based on a significance analysis of the variables on the neural-network output is examined. It is shown to have significant benefits in certain cases when compared.

Keywords:Decision trees,Hybrid Systems,Neural Networks

1. INDRODUCTION

DURING the last decade interest in artificial neural net- works has grown significantly, owing to their ability to represent nonlinear relationships that are difficult to model by means of other computational methods. Moreover, neural networks are easy to implement, are robust under the influence of noise, do not require *a priori* knowledge with regard to the distributions of data, and can be parallelized where rapid computation is critical.

However, but for the simplest structures, neural-network models are notoriously difficult to interpret. For example, the fact that neural networks have large degrees of freedom in the assignment of weights, can lead to a situation where two completely different sets of weights can yield nearly identical outputs. This drastically complicates the analysis and comparison of similar processes that are modeled or controlled by different neural networks. The opacity of neural networks can be seen as a major barrier to their implementation in a number of fields, such as medicine and engineering where mission critical applications demand a high degree of confidence in the behavior of relevant models. In order to overcome this limitation, various attempts have previously been made to extract rules from neural networks. Most of these techniques require special training methods and architectures for neural networks, or are based on assumptions that tend to restrict the ability of the neural network to generalize the underlying relationships in the data.

A more general algorithm not subject to these limitations is therefore proposed in this paper. More specifically, this algorithm does not depend on any assumptions with regard to the structure of the neural network or the input–output data and enables the characterization of the behavior of the neural network by means of a set of heuristic rules, similar to those obtained by means of other rule induction algorithms such as ID3 [1], [2], C4.5 [3], or CART [4]. Neural networks are generally better at approximating complex relationships for problems with predominantly continuous inputs. Therefore the rules extracted from the network not only clarify the neural-network model, but in some cases are also significantly more accurate than those derived by other machine learning methods, such as the aforementioned algorithms.

3.RESULTS

The results obtained with the various algorithms as specified previously are summarized in Tables I–III. The scores of Table I are in terms of the coefficient of determination [33], i.e., , where is the predicted value of the outcome, is the target value of the outcome, and y_t is the average target value of the outcomes. The y_{avg} corresponding fidelity to the neural network is given in Table II, while Table III contains the number of leaves in the decision trees, which is an indication of the complexity of the trees. For these binary trees the number of internal nodes is equal to one less than the number of leaves. For case study 1 the results reported on the ANN-DT algorithms in Table I are those with a noise factor of 0.3 and 1000 additional sample points to the neural network beyond the training points. Note that the additional points of the ANN-DT algorithm mentionedhere, are the sample points obtained using the neural network, as described in Section III-C, that is they are generated by the algorithm and not by (4). All the algorithms were therefore presented with the same training data, and were also evaluatedon the same test data.

Recalculate existing splits. Noise will make this task even harder. That is why CART's performance decreased further in the presence of noise, as can be observed from a plot ofthe algorithm's performance against the noise as shown in Fig. 5(a). Fig. 3(b) together with Fig. 5(b), which gives the respective number of rules obtained by each of the algorithms, reveal that the simpler trees generated by the ANN-DT(s)algorithm also perform better.

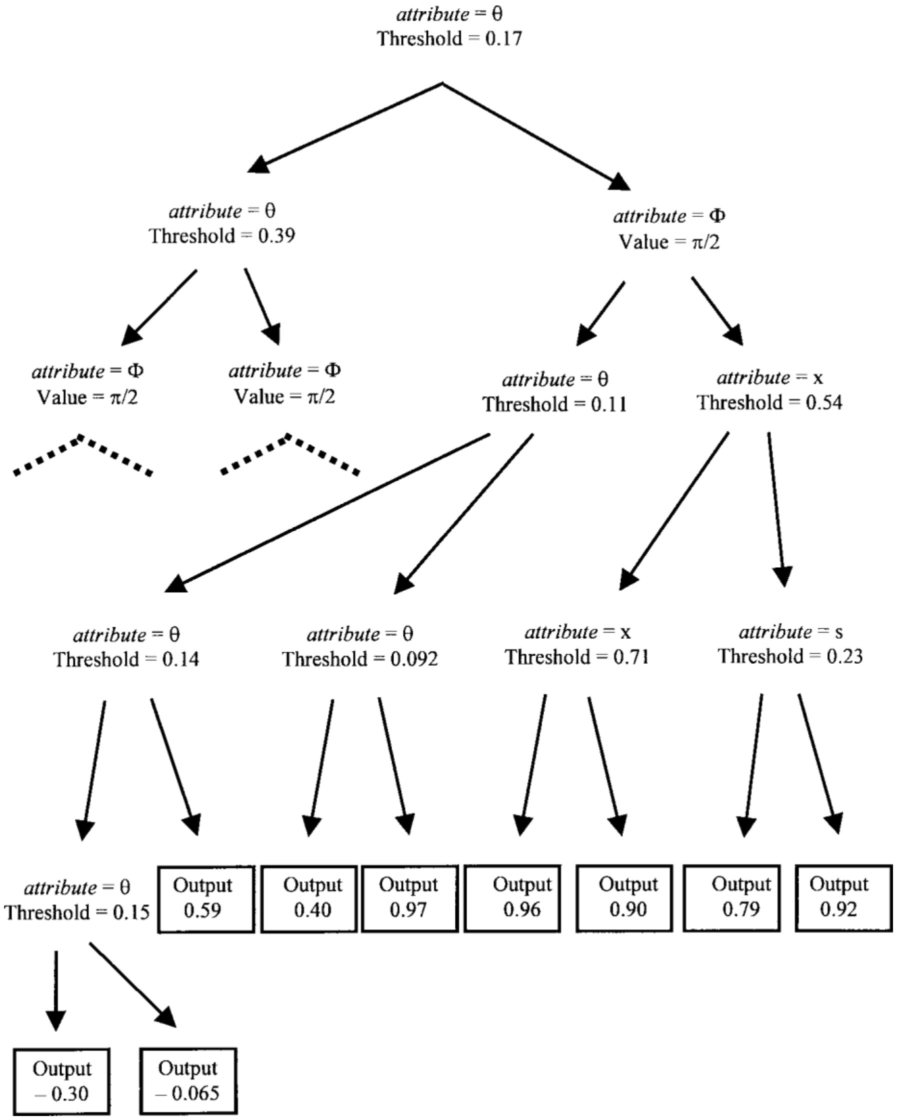

Fig. 4. The decision tree extracted by the ANN-DT(s) algorithm from the trained neural network for case study 1 with c = 0:3 using 1000 points to sample the neural network. If attribute a < Threshold the right subtree applies, else the left subtree is valid.

A. *Case Study 1: Sine and Cosine Curves*

The fidelity and accuracy of ANN-DT(s) and ANN-DT(e), given in terms of the coefficient of determination (R^2), for different numbers of neural network sample points, are shown in Fig. 3(a). The number of rules obtained by the respective algorithms for the test runs is given in Fig. 3(b). The results shown in Fig. 3(a) clearly suggest that the extra samplingpoints make a significant contribution. Both accuracy and tree size increased with an increase in the number of sample points. Although the ANN-DT(e) algorithm failed almost completely to capture the overall trends in the data, the ANN-DT(s) algorithm yielded satisfactory results, even with few sample points.

For as 0.45 and as 0.30 at the first split. The $\pi(f)\phi$ other two attributes each had a level of significance less than 0.02. It canbe seen from the decision tree in Fig. 4 that ϕ was selected once at tree depth of one and twice at a tree depth of two.

The greedy attribute selection measure of ANN-DT(e) and CART split relatively late on the attribute ϕ, because a split on this attribute caused very little immediate gain. Although not shown in the figure, CART split the data on this attribute once at a depth of two, three, and four and ANN-DT(e) split even lower at a depth of three, four, and five. After too many splits on insignificant attributes the data were too sparse to pick up any underlying trends.

This shortcoming of the greedy attribute selection measure that is used by CART cannot be compensated for at a later stage by pruning. Pruning will attempt to replace a subbranch by leaves, but will not

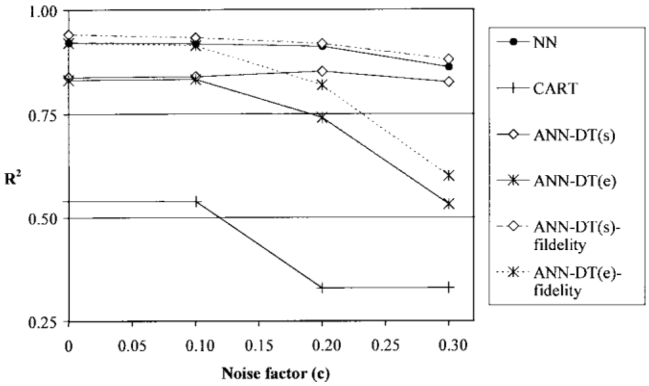

Fig. 5. (a) A plot of the accuracy (R^2) of the algorithms for different noise factors in the test data of case study 1. The dashed line indicates the fidelity with respect to the neural network, a multilayer perceptron from which the rules were extracted. The size of the data set with which the ANN-DT(s) and ANN-DT(e) algorithms sampled the neural network beyond the 300 training points, was held constant at 500. All the decision trees were pruned statistically with an value of 0.05. (b) The numbers of rules induced by the algorithms in case study 4.

4.**DISCUSSION**

In all the case studies, the ANN-DT algorithms successfully extracted faithful rule-based representations from the trained neural-network models. An interesting result is that the rules induced by the ANN-DT(e) algorithm are as accurate and sometimes more accurate than those induced by the CART algorithm. Similar results were found by Craven and Shavlik [19], who compared the TREPAN algorithm to classification trees induced by C4.5 [3] and ID2-3 [34]. The results indicate that for many problems inductive techniques, like C4.5 and CART, do not use all the information that is contained in the original data. A possible source of this loss of information is that the techniques split the data recursively into branches in such a way that the data to be processed in the underlying branches are isolated from another. This means that any trend that might exist between the input and the output data which is distributed over points belonging to different branches, will not be discovered by these algorithms. It also means that pointsnot complying with this trend as a result of noise cannot be identified and a rule can arise out of these exceptions that does not generalize well.

If it is assumed that the neural network detects these trends and does not overtrain on outliers in the data, both the ANN- DT(e) and ANN-DT(s), as well as the TREPAN algorithm are evolved on data where these exceptions are already removed. Moreover, the more densely sampled points help in finding better estimates of the threshold values at which the data should be split. In contrast, the C4.5 and CART algorithmscan only estimate this value to lie somewhere between two points of the subset of the original data that belongs to the branch in which the next split is to be made. This subset is much denser in the case of the algorithms extracting rules from neural networks.

Problems that extend over different branches of the tree, can be estimated in regions where there is very little or no training data. This is because the neural network does not split the dataand can extend such a decision boundary between the trainingpoints via interpolation.

The ANN-DT algorithm can samplein these regions and produce additional rules to cover these regions. For the same reasons the ANN-DT algorithms tend to maintain a higher fidelity with respect to the neural network. Both ANN-DT(s) and ANN-DT(e) can be applied to nonparametric models other than feedforward neural networks,without making any assumptions about the model’s internalstates or the nature of the data. The computational time of the ANN-DT algorithm scales linearly with the neural- network size and is only dependent on the time it takes theneural network to assign a label to a data point.

However, the algorithm’s computational time does suffer from the curse of dimensionality. In order to achieve a higher density of points than that of the training data, progressively more sample points are required as the dimensionality of the data increases. This problem can be reduced somewhat by initially using fewer sample points and growing the tree from a node in a best-first manner.

This is performed in the TREPAN algorithm [19] by presenting the node that is most likely to increase fidelity with sufficient samples. Naturally the number of these sample points also needs to grow exponentially with the dimensionality of the data in order to achieve the same accuracy, as once a split is made in the tree it cannot be adjusted later.

In case study 1, it was seen that the use of the significance analysis in attribute selection can hold significant advantages over the greedy variance criterion. θ Although single splits on attributes ϕ and did not cause a significant decrease in the normalized variation of the data in case study 1, changes in these attributes σ_f were nevertheless correlated with changes in the output of the neural network and therefore had high values. Provided that the neural network accurately models the input–output relationships represented by the data, the significance analysis therefore learns from the trained neural network which attributes have the most influence θ over the data set covered by a particular node. On the other hand, the greedy splitting criteria of the CART and ANN-DT(e) algorithms did not compensate for the periodicity of the function with respect to the attribute . The ANN-DT(e) algorithm could use additional sample points to obtain a satisfactory performance.

5.CONCLUSIONS

A novel approach has been developed to extract decision trees from trained feedforward neural networks, regardless of the structures of these networks. It was found that in some cases these rules were significantly more representative of the behavior of the neural network than rules extracted from the training data only.

Alternatively, the algorithm can be used as a method to extract rules from data sets. These rules appear to be of similar accuracy as those obtained with CART. In fact, in some cases a significant improvement could be obtained with the ANN-DT algorithm.

In one particular case it was demonstrated that the significance analysis of the ANN-DT(s) could correctly identify the most important attributes and build valid sets of rules. In contrast to this, a greedy error driven procedure, such as used in CART and ANN-DT(e), failed to identify the most important attributes. As a result, rules derived with CART and ANN-DT(e) were comparatively inaccurate, while the ANN- DT(e) algorithm could only find more accurate rules by using many more sample points than ANN-DT(s).

Unlike a sensitivity analysis that only considers the partial derivatives, the significance analysis proposed in this paper takes the correlational structure of the data into account. This significance analysis appears to be a suitable splitting criterion near the root of the decision tree, whereas a greedy splitting criterion would be better at splitting the lower branches of the tree.For some case studies, additional sampling of a trained neural network resulted in appreciable improvement in the accuracy of the rules extracted from the network.

REFERENCES

[1] R. Davis, B. G. Buchanan, and E. Shortliffe, “Production rules as a representation for a knowledge-based consultation program,” *Artificial Intell.*, vol. 8, no. 1, pp. 15–45, 1977.

[2] J. R. Quinlan, “Induction of decision trees,” *Machine Learning*, vol. 1, pp. 81–106, 1986.

[3] _____, *C4.5: Programs for Machine Learning.* San Mateo, CA: Mor- gan Kaufmann, 1993.

[4] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees.* New York: Chapman and Hall, 1984.

[5] R. Andrews, J. Diederich, and A. B. Tickle, “Survey and critique of techniques for extracting rules from trained artificial neural networks,” *Knowledge-Based Syst.*, vol. 8, no. 6, pp. 373–383, 1995.

[6] M. W. Craven and J. W. Shavlik, “Using sampling and queries to extract rules from trained neural networks,” in *Proc. 11th Int. Conf. Machine Learning,* San Francisco, CA, 1994.

[7] G. Towell and J. W. Shavlik, “Extracting refined rules from knowledge based neural networks,” *Machine Learning,* vol. 13, pp. 71–101, 1993.

[8] L. M. Fu, “Rule learning by searching on adapted nets,” in *Proc. 9th Nat. Conf. Artificial Intell.,* Anaheim, CA, 1991, pp. 590–595.

[9] S. I. Gallant, *Neural Network Learning and Expert Systems.* Cam- bridge, MA: MIT Press, 1993.

[10] I. K. Sethi, “Neural implementation of tree classifiers,” *IEEE Trans. Syst., Man, Cybern.,* vol. 25, pp. 1243–1249, 1995.

[11] S. Thrun, “Extracting provable correct rules from artificial neural networks,” Institut für Informatik III Universität, Bonn, Germany, Tech. Rep. IAI-TR-93-5, 1994.______, “Extracting rules from artificial neural networks with distributed representation,” in *Advances in Neural Information Processing Systems,* G. Tesauro, D. Touretzky, and T. Leen, Eds., 1995, vol. 7.

[12] E. Pop, R. Hayward, and J. Diederich, “RULENG: Extracting rules from a trained

artificial neural network by stepwise negation," in *QUT NRC*, Dec. 1994.

AUTHOR PROFILES

Dr.M.Rajaiah , Currently working as an Dean Academics & HOD in the department of CSE at ASCET (Autonomous), Gudur, Tirupathi(DT).He has published more than 35 papers in, Web of Science, Scopus Indexing, UGC Journals.

Dr.N.Krishna Kumar completed his Bachelor of Technology in Computer Science and Engineering.He completed his Masters of Technology in Computer Science and Engineering. Awarded Ph.D in Computer Science and Engineering at Pondicherry University (Central University), Puducherry. He has published more than12 papers in indexing Journals.Currently working as an Associate Professor in the department of CSE at ASCET (Autonomous), Gudur, Tirupathi(DT). His areas of interest include, Data Mining, Cloud Computing and MachineLearning.

Mr. Akula Sujan Kumar, as B.Tech student in the department of CSE at Audisankara College of Engineering and Technology, Gudur.

Mr.Amruthala Anil Kumar, as B.Tech student in the department of CSE at Audisankara College of Engineering and Technology, Gudur.

Mr.Kamineni Venkat Chowdary, as B.Tech student in the department of CSE at Audisankara College of Engineering and Technology, Gudur.

Ms. Addam Vennela, as B.Tech student in the department of CSE at Audisankara College of Engineering and Technology, Gudur.