Knowledge Extraction from Radial Basis Function Networks and Multi-layer Perceptrons

Kenneth J. McGarry, Stefan Wermter and John MacIntyre
School of Computing, Engineering and Technology,
St Peters Campus, St Peters way,
University of Sunderland, Sunderland, England, SR6 ODD
email: cs0kmc@cis.sunderland.ac.uk

1 Abstract

Recently there has been a lot of interest in the extraction of symbolic rules from neural networks. The work described in this paper is concerned with an evaluation and comparison of the accuracy and complexity of symbolic rules extracted from radial basis function networks and multi-layer perceptrons. Here we examine the ability of rule extraction algorithms to extract meaningful rules that describe the overall performance of a particular network. In addition, the research also highlights the suitability of a specific neural network architecture for particular classification problems. The research carried out on the extracted rule quality and complexity also has a direct bearing on the use of rule extraction algorithms for data mining and knowledge discovery.

2 Introduction

The work described in this paper is concerned with an evaluation of the accuracy and complexity of symbolic rules extracted from radial basis function (RBF) networks and multi-layer perceptrons (MLP). RBF neural networks [5] and MLP networks [4] are two of the most widely used neural network architectures. RBF networks are a localist type of learning technique [3]. Local learning systems generally contain elements that are responsive to only a limited section of the input space. This may entail separate storage in memory for each pattern unless the representational elements are able to cover (as in the case of RBF hidden units) a given area around the input pattern.

This is quite different from the distributed approach of MLP networks. MLP's are able to store many patterns within a limited memory, i.e. the learned patterns are stored across all weights and thresholds. This property is known as superposition and enables the efficient storage and recall of individual patterns. However, both types of networks are good at pattern recognition and are robust classifiers, with the ability to generalize in making decisions about imprecise input data. They offer robust solutions to a variety of classification problems such as speech, character and signal recognition, as well as functional prediction and system modeling where the physical processes are not understood or are highly complex. The main difference is that RBF networks may require more hidden units than MLP's to represent the same data set.

The local nature of RBF networks makes them a suitable platform for performing rule extraction. Here we examine the ability of rule extraction algorithms to extract meaningful rules that describe the overall performance of a particular network. The research carried out on the extracted rule quality and complexity also has a direct bearing on the use of rule extraction algorithms for data mining and knowledge discovery. Rule extraction is recognized as a powerful technique for neuro-symbolic integration within hybrid systems [15; 8].

To illustrate how different classifiers can partition the data space and thereby produce varying accuracies, figure 1 shows the decision boundaries for a RBF and a MLP network on a two class problem.

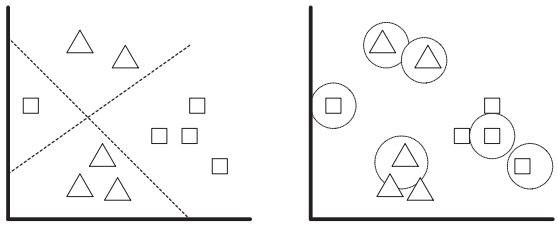

Figure 1: Class decision boundaries

The MLP uses one or more hyperplanes to isolate the classes. Hyperplanes may be positioned anywhere in input space and can extend infinitely which leads to extrapolation problems and causes complications during rule extraction. The RBF network uses a local approach

which effects only a specific data point and perhaps a small number of other points depending on the basis function width.

The adjustable parameters within a radial basis function network that effect classification accuracy and that may provide information for rule-extraction are the number of basis functions used, location of the centre of the basis function, width of the basis function, and the weights connecting the hidden RBF units to the linear output units. Extracting rules from MLP networks is dependent on the number of hidden units, the weight and threshold values and the complexity of the learned target function.

This paper is structured as follows: Section three outlines the techniques used for rule-extraction from both neural network types. Section four describes the experimental results. Section five discusses the conclusions.

3 Rule Extraction from Neural Networks

In this section we discuss motivations, techniques and methodology for rule-extraction. The advantages of extracting rules from neural networks will be discussed in general terms applicable to most neural networks [1; 12].

- The knowledge learned by a neural network is generally difficult to understand by humans. The provision of a mechanism that can interpret the networks input/output mappings in the form of rules would be very useful.

- Deficiencies in the original training set may be identified, thus the generalization of the network may be improved by the addition/enhancement of new classes. The identification of noisy training data for removal would also enhance network performance.

- Analysis of previously unknown relationships in the data. This feature has a huge potential for knowledge discovery/data mining and possibilities may exist for scientific induction.

Rule extraction has been carried out upon a variety of neural network types such as multi-layer perceptrons [11; 6], radial basis networks [13], Kohonen networks [14] and recurrent networks [10].

Rule extraction may be viewed in one of two ways, first it can be seen as a technique for determining how the neural network performs any given input to output mapping. Second, often the rule extraction process may produce rules that are more accurate than the original neural network. In the second case the extracted rules may no longer provide a faithful reproduction of the original networks operation. However, this loss of fidelity is compensated for by an increase in classifier accuracy. Work by Fu has given insights into how this phenomena occurs [7].

3.1 RBF rule extraction by RULEX

The algorithm implemented for the extraction of rules from RBF networks is similar to the RULEX algorithm [2]. The local nature of each RBF hidden unit enables a simple translation into a single rule.

$$\begin{aligned} &IF\ Feature_1\ is\ TRUE\ AND \\ &IF\ Feature_2\ is\ TRUE\ AND \\ &IF\ Feature_n\ is\ TRUE \\ &THEN\ Class_x \end{aligned} \tag{1}$$

where a Feature is composed of upper and lower bounds calculated by the RBF centre μ_n positions, RBF width σ and feature "steepness" S. The value of the steepness was discovered empirically to be about 0.6 and is related to the value of the width parameter. The values of μ and σ are determined by the RBF training algorithm.

$$X_{upper} = \mu_i + \sigma_i - \mathrm{S} \tag{2}$$

$$X_{lower} = \mu_i - \sigma_i + \mathrm{S} \tag{3}$$

where:
μ = n-dimensional centre location
σ = width of receptive field

Input:
Hidden weights μ (centre positions)
Gaussian radius spread σ
Steepness S
Output:
One rule per hidden unit
Procedure:
Train RBF network on data set
For each hidden unit
For each μ_i
$X_{lower} = \mu_i - \sigma_i + \mathrm{S}$
$X_{upper} = \mu_i + \sigma_i - \mathrm{S}$
Build rule by:
antecedent = $[X_{lower}; X_{upper}]$
Join antecedents with AND
Add Class label
Write rule to file

Figure 2: RBF rule-extraction algorithm

3.2 MLP rule extraction by VIA

Validity interval analysis (VIA) extracts propositional IF..THEN type rules from pre-trained feedforward,

multi-layer perceptron (MLP) networks [11]. It uses the network parameters i.e. the weights and threshold values in conjunction with constraining values at the input and output units. VIA is a general purpose algorithm that assumes that the network consists of a feedforward architecture with continuous activation functions. VIA is based on the propagation of intervals of min-max values through monotonic real-valued functions [9]. The intervals specify the valid range of activation values a particular neuron may take. Although each hidden unit undergoes the VIA process individually, the extracted rules are based on the networks overall input to output mapping response. Some rule extraction techniques decompose a network into a number of sub-networks and merge the extracted rules after pruning the network architecture.

The VIA algorithm consists of two phases: a forward phase whereby interval constraints are propagated through the network, and a backward phase where the initial intervals are refined within tighter limits. The propagation of intervals during the backward phase is accomplished by using the simplex algorithm which is a linear programming technique.

The original intervals are refined by propagating them backwards through the network. Thrun viewed the problem of refinement as a linear programming exercise. This allows the arbitrary linear constraints to be incorporated into the calculation of the validity intervals. The backward propagation of activation intervals allows the calculation of tighter validity intervals. The whole process can therefore detect general conditions upon the output units i.e. more maximally generally rules than would be the case with only forward propagation. The Simplex algorithm is used to refine the initial intervals, constraints are placed upon these intervals i.e. one input is changed while the others are held constant. The Simplex is fed with this data and the routine should converge proving the changed interval is consistent with the others. Otherwise, a contradiction is generated because the new interval is not consistent with the networks weights and biases. This means that a lower bound has exceeded its upper bound.

4 Experimental Results

The data sets we used comprised a benchmarking data set, namely, the exclusive-or (XOR) dataset and Fishers's iris data set. The XOR dataset is a linearally inseparable, two class problem. However, to convert this problem from a Boolean to a continuous domain we added noise to the XOR dataset to produce 400 patterns. The iris data set consists of three classes of flowers with 50 patterns each. One class is linearly separable while the other two are not. Figure 5 shows the results of the rule extraction process in terms of number, accuracy and domain coverage of the rules. The coverage of the rules is based upon their accuracy in describing the operation of the neural network. Also the test results for the original neural networks are given.

Figure 3 is an example of a rule extracted from an RBF network trained on the Iris dataset. The antecedents consist of upper and lower bounds that must be present for the rule to be correct. The antecedent names describe the iris features, where SL and SW refer to sepal length and sepal width. PL and PW refer to petal length and petal width. Figure 4 shows a rule extracted from an

---

Rule 1
IF (SL $\geq$ 6.87 AND $\leq$ 7.3) AND
IF (SW $\geq$ 2.77 AND $\leq$ 3.22) AND
IF (PL $\geq$ 5.67 AND $\leq$ 6.12) AND
IF (PW $\geq$ 1.87 AND $\leq$ 2.32)
THEN..Virginica

Figure 3: Rule extracted from RBF network

MLP network trained on the Iris dataset. These rules are similar to the RBF rules since they both consist of upper and lower bounds. The original boundaries discovered by VIA.

Rule 1
IF SL[0.62 - 10.00] AND
IF SW[0.00 - 7.69] AND
IF PL[0.71 - 2.52] AND
IF PW[1.05 - 4.92]
THEN VIRGINICA[0.80 - 1.00]
THEN SETOSA[0.00 - 0.00]
THEN VERSACOLOR[0.010 - 0.02]

Figure 4: Rule extracted from MLP network

The RULEX approach produced reasonably accurate rules that were faithful to the original networks operation. The number of rules generated is based on the number of RBF units present, therefore the more complex dataset will tend to produce larger networks and hence more rules. However, the network architecture can be used to anticipate the number of extracted rules.

Using VIA to refine the intervals e.g. on the XOR dataset, four rules were derived. Many more were generated but with VIA it is possible to determine the most generally maximum rules. As with rules extracted from RBF networks, increased dataset complexity produces more rules. However, because of their distributed representation MLP networks require fewer hidden units. This means that the network architecture cannot be used as an indication of the number of potential rules to be extracted.

Figure 5: Results of classifier accuracy on data sets

Classifier	Number of rules Iris	Number of rules XOR	Accuracy (%) Iris	Accuracy (%) XOR	Rule coverage XOR/Iris(%)	Hidden units XOR/Iris
RBF network	–	–	98	100	–	20/53
MLP network	–	–	98	100	–	2/2
RBF rules	53	20	100	100	100/100	–
MLP rules	85	4	80	96	100/80	–

5 Conclusions

It is clear from the experimental work that more Iris rules are extracted from MLP networks than RBF networks. However, the XOR dataset having a simple and regular structure required fewer rules when represented by an MLP. The number of extracted rules can be very large for those MLP networks that have learned a complex mapping function. The actual coverage of the input space becomes very difficult and is reliant on the test-and-generate process for maximum coverage. It is likely that a number of rules describing nonlinear class features near hyperplane boundaries will be missed. This aspect will become more difficult with increasing dataset complexity. One facet of MLP rule extraction not investigated here would be to discover the effect of varying the number of hidden units within the MLP network.

Since each RBF unit compiles into a single rule, the rule extraction process is guaranteed to obtain all valid rules. The complexity and size of the rule set is therefore based directly on the number of RBF units within the network. The number of RBF units is determined by the training algorithm. The advantage of extracting rules from RBF networks is the certainty that the entire input space of the original network is covered. However, since RBF networks represent a local solution the extracted rules may not reflect the overall trend of the data set. The main advantage in extracting rules from RBF networks over MLP networks is the simplicity, accuracy and efficiency of the extraction algorithm.

6 Acknowledgments

The authors are grateful to Frederic Maire of Queensland University of Technology for providing the VIA source code used in the experimental work.

References

[1] R. Andrews, J. Diederich, and A.Tickle. A survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, 8(6):373–389, 1995.

[2] R. Andrews and S. Geva. RULEX and CEBP networks as the basis for a rule refinement system. In J. Hallam et al, editor, *Hybrid Problems, Hybrid Solutions*. IOS Press, 1995.

[3] C. G. Atkeson, A. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, pages 11–73, Feb 1997.

[4] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[5] D. S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, pages 321–355, 1988.

[6] T. Corbett-Clarke and L. Tarassenko. A principled framework and technique for rule extraction from multilayer perceptrons. In *IEE, Proceedings of the 5th International Conference on Artificial Neural Networks*, pages 233–238, Cambridge, England, July 1997.

[7] L. Fu. Learning capacity and sample complexity on expert networks. *IEEE Transactions on Neural Networks*, 7(6):1517–1520, 1996.

[8] K. J. McGarry, S. Wermter, and J. MacIntyre. Hybrid neural systems: from simple coupling to fully integrated neural networks. *Neural Computing Surveys*, 2(1), 1999.

[9] R. Moore. *Interval Analysis*. Prentice Hall, New Jersey, 1966.

[10] C. W. Omlin and C. L. Giles. Extraction and insertion of symbolic information in recurrent neural networks. In V.Honavar and L.Uhr, editors, *Artificial Intelligence and Neural Networks:Steps Towards principled Integration*, pages 271–299. Academic Press, San Diego, 1994.

[11] S. Thrun. Extracting rules from artificial neural networks with distributed representations. In G.Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems 7*. MIT Press, San Mateo, CA, 1995.

[12] A. Tickle, F. Maire, and J. Diederich. Extracting the knowledge embedded with trained artificial neural network: defining the agenda. In S. Wermter and R. Sun, editors, *Hybrid Neural Symbolic Integration Workshop, Neural Information Processing Systems*. Breckenridge, Colorado, 1998.

[13] V. Tresp, J. Hollatz, and S. Ahmad. Representing probabilistic rules with networks of gaussian basis functions. *Machine Learning*, 27:173–200, 1997.

[14] A. Ultsch, R. Mantyk, and G. Halmans. Connectionist knowledge acquisition tool: CONKAT. In J. Hand, editor, *Artificial Intelligence Frontiers in Statistics: AI and statistics III*, pages 256–263. Chapman and Hall, 1993.

[15] S. Wermter and R. Sun. *Hybrid Neural Symbolic Systems*. Springer, Heidelberg, 1999 (to appear).