Comparison of Extracted Rules from Multiple Networks

Edwin Che Yiu CHOI and Tamás Domonkos GEDEON

School of Computer Science and Engineering
The University of New South Wales
Sydney 2052 AUSTRALIA
Fax: +61 2 385 5995
E-mail: { edwin | tom }@cse.unsw.edu.au

ABSTRACT

Neural networks can be trained to provide solutions in application domains where clear rules which would allow symbolic solutions do not exist. Neural networks in these domains still suffer from a major disadvantage, in that there is no explanation for why a particular decision was made by the network. We have generalised on our previous work on generating explanations for trained back-propagation neural networks to extract rules.

We have found that there is significant variation in the quality of rules extracted from networks which have not been tuned for the task, and that the neural network correctness on the test set is not well correlated with the often better correctness of the extracted rules on the test set.

1. Assumptions

In this paper we assume a feed-forward network of three layers of processing units. All connections are from units in one level to the subsequent one, with no lateral, backward or multilayer connections. Each unit has a simple weighted connection from each unit in the layer above. The network is trained using a training set of patterns with desired outputs, using back-propagation of error measures [5]. In the examples we cite here we have used the basic logistic activation function $y=(1+e^{-x})^{-1}$.

2. Introduction

Feed-forward networks of a few layers trained by back-propagation can be used to solve a variety of problems. Training by back-propagation is popular because of its simplicity theoretically, and the ease of use and production of such networks, and the wide availability of low cost simulators.

The ability to learn from examples makes these neural networks useful and powerful tools. One application that is becoming increasingly popular is that of using the trained neural network in the role of a human expert. This has the advantage of reducing large amounts of expensive expert time by learning from example data during training. This method has obvious advantages, however unlike conventional expert systems the neural network has no ability to provide any rule trace, or some sort of explanation as to how it comes to its conclusions. Rather than storing sets of rules the knowledge contained in a neural network is stored in the weight values distributed throughout the entire network.
It is well known that to achieve the best generalisation we must stop training back-propagation neural networks when the error on the test set is minimum. A concomitant of this observation is that using random lengths of training times would produce data leading to conclusions such as that the success on the training set is not well correlated with success on the test set. That is, an overtrained network will have very good training set performance but low test set performance, an under-trained network will have low performance on either set, and so on. We will use this observation in the results section.

3. Network Topology

The networks analysed were identical in all parameters except for the initial (and hence final) weights, and match the networks used in our previous work on this data set [2,8].

The topology is a three layered neural network consisting of fourteen inputs, five hidden units and four output units.

4. Application Domain

The data consists of a set of assessment marks in an undergraduate Computer Science subject, with the goal to predict the final result that a student will receive. The assessments included in the data set combine to yield only 40% of the total mark, the remaining 60% being the final examination mark

which is omitted. Thus, the task is to predict the final mark which is normally calculated using the exam mark, from only the 40% weight of class assessments during the session.
The classification of marks fall into the following categories:

- Distinction or above, being a mark of 75 or greater, represented by output 1.
- Credit, being a mark between 65 and 74, represented by output 2.
- Pass, being a mark between 50 and 64, represented by output 3.
- Fail, being a mark less than 50, represented by output 4.

5. Summary of Previous Work

We have produced an explanation facility for back-propagation trained neural networks [2, 8]. Explanations do not simply consist of a set of rules (or rule traces) as to why the network came to its conclusion, but also include identification of important factors in the input, and the next most likely output of the network.

The Causal Index C_{ki} is the rate of change of k with respect to i . This indicates the relationship between the k^{th} output and the i^{th} input neurons in the trained neural network. The value of C_{ki} indicates a positive or negative correlation between input and output signals. Using this value, explanation of the importance of each input could be given by using equations such as: If C_{ki} is Positive and Large then 'If i is large then k is large'.

Unlike [9x, 3], we have found that the assumption they make that the product $f'(U_{k2}).f'(U_{j1})$ is constant for all k and j does not hold for the domains we have tried. This is unfortunate, as otherwise the influence of x_i on y_k could be statically determined from the weight matrix of the trained network.

To compensate for this problem, we have introduced our notion of *Characteristic Patterns* as follows. The formula used in calculating the Causal Index causes one major problem: the results are input specific. This is a problem in two ways. If the analysis can not be generalised to satisfy any set of possible inputs or even any arbitrary subset of the set of possible inputs, then creating explanations for each input pattern is necessary. This is firstly not very efficient, and secondly such extremely specific explanations are not ideal. That is, an explanation is at least implicitly something humans can generalise from. Thus, we require a more general explanation which focuses on the key elements distinguishing particular input patterns.

This problem is overcome by using input values representative of the input set. Finding a single input pattern that is representative of the entire input set is impossible. To achieve this an input pattern must be found representing all the patterns that cause an output to be both on and off; an obvious contradiction. To solve this problem input patterns are split into classes according to their effect on an output. When the input pattern causes the output being analysed to be turned on, it is classed as an *ON* input pattern, otherwise it is classed as an *OFF* input pattern.

Using some statistical or other means [4] on each input value, a pattern representing each input class is created as a *Characteristic* pattern for that class. A characteristic *ON* pattern is a pattern characteristic of those input patterns which turn an output on. Similarly a characteristic *OFF* pattern is a pattern characteristic of those input patterns which turn an output off. Note that we consider here networks where the output units represent membership in categories by high output values. For networks where an output unit's value is used directly and not just as a classification threshold, we can define *Characteristic* patterns over subranges of the value of the output unit activation.

In our previous work, by using the causal index on characteristic input patterns, we produced lists of inputs which were significant in reaching the decision made, a set of rules governing this decision, and the next most likely decision the network could have made. This method correctly explained 94% of the decisions made by a sample network.

6. Results of Current Work

We have trained a number of neural networks with identical topology but different initial weights. The networks were trained until the error on the test set was minimum, and the rules extracted.

We highlight here three sample runs, displayed in Table 1, selected as follows.

The first sample is run A, which accords with our initial expectations, in that the extracted rules are slightly worse than the neural networks itself on the test set. We expected this result because in our previous work on extracting explanations for the results of trained networks we could produce explanations for some 94% of network decisions. Explanations must match the network conclusion even if they do not match the known desired outputs of a test set, and are different from the rules we

generate here, in that here we focus on the harder task of extracting rules which are correct with respect to the desired outputs in spite of the neural network.
The second sample is run D, where the rules extracted are least useful, with the fewest patterns in the test or training sets identified correctly by the rules.

The third sample is run H, where the neural network performance on the test set is significantly lower than either of the other two samples A and D, but the rules extracted produce the best performance (64%) on the patterns in the test set.

	training set %		test set %	
	net	rule	net	rule
Run A	**83.0**	**70.0**	**62.3**	**56.6**
Run B	87.0	73.0	62.3	64.1
Run C	66.0	69.0	43.3	60.4
Run D	**90.0**	**59.0**	**62.3**	**45.3**
Run E	80.0	59.0	60.4	50.9
Run F	74.0	71.0	41.5	52.8
Run G	66.0	75.0	56.6	64.1
Run H	**67.0**	**75.0**	**43.4**	**64.1**
Run I	67.0	73.0	54.7	60.4
Run J	73.0	75.0	50.9	64.1
ave. A–J	75.3	69.9	53.8	58.8
Run TG	75.0	70.0	66.0	58.8

Table 1. Percentage correctness of methods

The most common result was of the kind demonstrated by run H, with slightly lower than average neural network performance on the test set, but with more correct performance of the extracted rules on the same test set.

Samples A and D seem to belong to a group where the average neural network performance on the test set is higher than average, but the performance of the extracted rules on the same test set is worse than the performance of the neural network. This is particularly true for run D, which has the best aggregate or singular network performance on the training and test sets, and the worst extracted rule aggregate or singular performance on the training and test sets.

Note that the final row in Table 1 entitled "Run TG" is the results using our previous work with a network carefully tuned to maximise the performance over the test set patterns. The results quoted here differs from [2,8] in that these figures are for rules rather than the simpler explanation generation task.

Furthermore, the training set used to produce the weights was reduced in size from 100 to 86 patterns to remove outliers [6]. It is particularly interesting to note that the extra work involved in this process of tuning has only improved the neural network performance on the test set, and has provided no significant benefit in comparison to the average result of our rule extraction on our untuned networks.

Sample extracted rules are shown in Table 2.

For Fail using Own Con:
(H2 <= 0.96) AND (Enrolment >= 0.13)
For Fail using Con for Pass :
(H2 <= 0.01)

Table 2. Category *Fail* rules for Run G.

The interpretation of the rules is in the context of the appropriate characteristic patterns. Thus, the rule for a Fail using its own Characteristic ON pattern contains the extra conditions that a pattern must fulfil to be a Fail if the most similar characteristic pattern by Euclidean distance was the Characteristic ON pattern for Fail. That is, a pattern which looks like an archetypal fail is a fail if it also fulfils the rule. The second rule shown in Table 2 is the only other rule producing Fail results. A pattern which looks like an archetypal Pass, but was very low on input H2, is actually a Fail.

These two rules can be justified from an educational viewpoint based on the raw data. A student whose performance was overall of a Fail standard, but did extremely well on assignment H2 obviously is improving their understanding just before the final examination. Similarly, a student with an overall passing grade with an result in H2 so low as to indicate probably the omission of the assignment suggests that the student has stopped trying.

7. Discussion

If we naïvely trained a number of neural networks and terminated training at arbitrary times, we would be able to identify a number of categories and come to some conclusions about the performance on the training set versus performance of the test set. We might conclude that high performance on the training set correlates badly with performance on the test set. We would then wonder if there was some measure we could identify which would allow us to maximise the performance on the test set directly. This example is obvious as we can simply discover the point at which the test error is minimised using cross-validation.

There is some analogy with the naïve notion proposed in the above paragraph and our observations with respect to the neural network versus extracted rule performance on the test set. That is, we postulate that there is some optimum point to stop training the neural network to produce the best generalised rules on the test set.

The fact that we can extract rules which perform better on the test set than the neural network itself suggests that the neural network was in some sense overfitting the data. That is, since the production of rules requires the discarding of some information, if we observe a consistent improvement in performance over a number of trials this means that the information being discarded was not necessary.

The above observation contradicts a long established assumption in the neural network community, that we can quickly achieve the same ends by training a "just right" network and a "too large" network so long as we terminate training by cross-validation. We define a "just right" network conventionally as one that can account for a test set at a desired level of accuracy, but the removal of any free parameters in the form of one or more hidden neurons will degrade performance unacceptably.

8. Conclusions And Future Work

We have shown that there is significant variation in the quality of rules extracted, with out method producing better results on the test set than the neural network itself, and that the neural network correctness on the test set is not well correlated with the correctness of the extracted rules on the test set.

Future work will be required to investigate whether training the neural network which produce results of the form of A and D should be trained more, or less, to improve the correctness of the rules generated, and some means to identify the properties which will lead to cases such as run H identified.

Further work will also be required to investigate whether the property we have observed is a feature of our own method or common to other rule extraction methods. We believe based on related experience in analysing the relationship of weight matrices to neural network behaviour [1] that this likely to be common to other methods also.

Finally, the quality of rules extracted from just right versus large networks with quick termination of training may provide an indication of whether there are qualitative differences between these two approaches to completing the training of networks.

9. References

]1] Gedeon, TD "Indicators of Hidden Neuron Functionality: The Weight Matrix versus Neuron Behaviour," *Proceedings ANNES International Conference*, 4 pages, Canterbury New Zealand, 1995.

[2] Gedeon, TD and Turner, H "Explaining student grades predicted by a neural network," *Proceedings International Joint Conference on Neural Networks*, pp. 609-612, Nagoya, 1993.

[3] Hora, N, Enbutsu, I and Baba,K "Fuzzy rule extraction from a multilayer neural net," *Proceedings IEEE*, vol. 2, pp. 461-465, 1991.

[4] Nejad, AF and Gedeon, TD "BiDirectional Neural Networks: Class Prototypes," *Proceedings IEEE International Conference on Neural Networks*, 6 pages, Perth, 1995.

[5] Rumelhart, DE, Hinton, GE, Williams, RJ, "Learning internal representations by error propagation," in Rumelhart, DE, McClelland, *Parallel distributed processing*, Vol. 1, MIT Press, 1986.

[6] Slade, P and Gedeon, TD "Bimodal Distribution Removal," in Mira, J, Cabestany, J and Prieto, A, *New Trends in Neural Computation*, pp. 249-254, Springer Verlag, Lecture Notes in Computer Science, vol. 686, 1993.

[7] Tan, A-H "Integrating Rules and Neural Computation," *Proceedings IEEE International Conference on Neural Networks*, 6 pages, Perth, 1995.

[8] Turner, H and Gedeon, TD "Extracting Meaning from Neural Networks," *Proceedings 13th International Conference on AI*, vol. 1, pp. 243-252, Avignon, 1993.

[9] Yoda, M, Baba, K and Enbutu, I "Explicit representation of knowledge aquired from plant historical data using neural networks," *International Joint Conference Neural Networks*, San Diego, vol. 3, pp. 155-160, 1991.