Proceedings of 1993 International Joint Conference on Neural Networks

Explaining student grades predicted by a neural network

T. D. Gedeon and **H. S. Turner**

School of Computer Science & Engineering
The University of New South Wales
P.O. Box 1, Kensington 2033
AUSTRALIA

Abstract

We have trained a back-propagation trained feed-forward neural network to predict student performance in a large undergraduate Computer Science subject at the University of New south Wales. The prediction uses partial grades from during the teaching session to predict the final grade. The exam mark which is the major component (60%) of the overall grade is not used.

The purpose of this network is to allow students to predict the final grade they are likely to achieve based on current performance, and obviously to improve their performance if the predicted grade is below their expectations. By itself, however, the network is not adequate as it provides no feedback as to why their performance merits a particular grade. We therefore generate an explanation of the conclusion reached by the neural network for predicting particular student grades.

Introduction

The experiments were performed on the full set of 153 patterns being the class results of an undergraduate Computer Science subject COMP1111 at the University of New South Wales.

The raw data consisted of the results from a number of laboratory exercises, assignments and a mid-term quiz all of which compose 40% of a student's mark for the subject. The exam mark which comprises the remaining 60% has been omitted, and the final aggregate mark is provided.

The goal of the exercise is to predict the final mark based on the partial marks. The educational imperative for such prediction is to be able to provide for students a reliable prediction of their final mark based on their current performance. This will of course be expected to invalidate the prediction in that students with low predicted final marks will take extra steps to improve their preformance. A sample of the raw data is shown in Table 1.

COMP1111 More Computing Sorted on student ID 18 May 92 10:34:12 Page 1

Regno	Crse/Prog	S	ES	Tutgroup	lab2		lab4		h2		p1		mid		final	
						tutass		h1		lab7		f1		lab10		
					3	5	3	20	20	3	20	20	45	3	100	
0275000	3400		1 F	T10-yh	2.5	3	3	18	4.5	3	14	18.5	24	2.5	68	
0275105	3420		1 F	T9-ko	3	4	2.5	17	17	3	5	14	10	2.4	56	
0275139	3420		1 F	T4-ko	0	5	2.5	18	17	3	6	10	28	2.4	57	
0275164	3400		1 F	T2-no	.	3	1.5	8.5	.	1.5	.	.	10.2	2.4	44	
0275279	3420		1 F	T2-no	3	3	.	19	18	2	5.5	4	20	2.4	60	
0275282	3400		1 F	T4-ko	2.5	3	3	19	.	3	.	10	16	2.4	51	
0275298	3400		1 F	T9-ko	3	5	2.5	17	18	3	8.5	18	21	2.4	61	
0275315	3420		1 F	T10-yh	2	3	0.5	14	.	1	.	.	7	2	26	
0275567	3400		1 F	T10-yh	.	3.5	2.5	19.5	.	2.5	.	.	11.5	.	36	

Table 1: Raw Data

This data has also been used previously to determine the effect of reducing the size of the training set on network performance on a test set [1, 2].

Explanation mechanisms

A number of approaches have been used to attempt to explain neural network conclusions, which at minimum require the production of a set of rules expressing the knowledge learnt by the neural network.

The simplest solutions depend on pruning [3, 4] the network to a minimal size, forcing the internal representation to approximate a symbolic and readily extractable mode. This extreme pruning does however reduce the robustness of the neural network solution [5], so other methods are required.

Rules can be simply generated by effectively encoding an existing network and its activation values for a particular case into rules, however even for a single neuron, the number of IF-THEN rules can grow exponentially with the number of inputs [7]. Logical rule derivation [6] classifies neurons into boolean or linear classes depending on where on the sigmoid their activation values lie, and can produce reasonably compact rule sets.

Beyond producing rules, a full explanation mechanism needs to provide alternatives answers, and variations in the context which can lead to different conclusions.

There are two major approaches which have such contextual properties. Sensitivity Analysis [8] is a simple method of finding the effect an input has on the output of the network. The relationship of an input neuron i and an output neuron k is found by determining the impact that a small change in i has on k. If drastic change occurs i is considered to be one of the key factors in producing the current activation value of k. This is of course quite computationally expensive. Alternatively, the differentiability of the activation function can be used to mathematically derive the rate of change of k with respect to i. This *causal index* can be used to generate rules [9, 10]. It has been shown that a common simplifying assumption which allows the use of the static weight matrix of the neural network does not hold in general, and that the full causal index must be used [11].

Thus, we use a full causal index method. This would be computationally almost as expensive as sensitivity analysis, thus we use a compressed representation of the training set to generate most explanations, and only generate explanations for specific cases where necessary. This compressed representation is in the form of characteristic input patterns.

Characteristic Input Method

Input patterns are classified in terms of their effect on each particular output. The set of patterns which turn that output on are used to produce a *characteristic ON* pattern. This can be done by a number of statistical methods. In this work we have used the arithmetical mean of the vector components. Similarly, *characteristic OFF* patterns are found. In work to be reported elsewhere using medical diagnosis cases, the patterns closest to the centres of clusters of such patterns are used as cornerstone characteristic cases. In that domain, such cases are particularly useful in the medical acceptability of explanations.

The use of *ON* and *OFF* above do not indicate that our method is binary in nature, rather that for this application this level of discrimination was sufficient. Thus, we can very simply use characteristic patterns tuned to a number of subranges of values of each output neuron for instance.

Explanation Procedure

To produce concise, understandable explanations our facility follows the methodology:

1. Liken the input pattern to the characteristic input patterns, and present the most similar to the user.
2. In addition present inputs considered 'important' for the current network output, and their values in the characteristic pattern.
3. Produce a set of rules, and evaluate to confirm accuracy.
4. Give the network's next most likely output.

The first section of the explanation methodology can be compared to some forms of explanations used by human experts to explain the results they obtain. As an example consider a doctor explaining why he has come to a particular diagnosis. A typical explanation may include statements such as "You have all the classic symptoms of X." Presenting the characteristic input pattern is similar to this kind of behaviour.

As there may be more than one characteristic input pattern produced from a network's training set (one for each distinctive output) the input pattern is compared to all these patterns and the most similar characteristic pattern is presented.

Once the correct characteristic input pattern has been found it is a simple operation to present the inputs important in the current input pattern. The inputs appearing in the graph of this pattern are presented to the user, with their characteristic values. This is done even in cases in which the pattern is being used as a characteristic *OFF* pattern for another output. This procedure can be likened to the manner a doctor explains a diagnosis he has made. The patient has most of the standard symptoms of a certain disease X, however, one symptom of extraordinary proportion leads the doctor to the conclusion that the diagnosis is disease Y.

In some cases (such as that described above), patterns are similar to that of one characteristic pattern, but result in a different output. In these cases, rules are presented to provide an invaluable insight into how the network is making its decisions. In other cases the rules produced can offer some help in understanding the network's actions.

To select the next most likely output, the simple comparison used in the choice of characteristic inputs is again used. In this case however, the next most likely output is that whose characteristic *ON* input pattern is most similar to the current input pattern, other than the characteristic input pattern for the network's current output. This method produces the output (other than the current output) that will occur by making the smallest possible change to the input pattern.

Examples of rules generated

The rules generated for a Distinction using characteristic patterns were:

Characteristic pattern	Rule Set
ON Distinction	(h2 ≤ 0.52) AND (mid ≤ 0.98)
ON Credit	(Crse ≥ 0.92) AND (Stage ≤ 0.93) AND (lab2 ≥ 0.39) AND (Midterm ≥ 0.12) AND (lab10 ≥ 0.14)
ON Pass	mid ≥ 0.85

Surprisingly, there was a negative correlation between a good mark for assignment *h2* and a final grade of Distinction. Upon reflection, this is plausible. That assignment is the last assignment at the end of session, and is quite difficult, but worth few marks relative to the final exam. We have now simplified this assignment and reduced its value to avoid 'misleading' students into spending too much time on it. Similarly, getting one of the top marks in the midterm quiz (and thus producing a scaled value close to *1*) probably leads to complacency.

The *ON Credit* and *ON Pass* rules need some explanation. These rules are for the characteristic patterns for *Credit* and *Pass* respectively, but changes to single input values change the result to a Distinction.

The *ON Credit* rules provide some useful information. The course code is straightforward, the codes indicate the student's major, with Science and Engineering course codes being numerically after those for Arts for example. We had not expected a major effect from the course of study and encoded the three main course codes together with a miscellany of other codes into a single input variable. The rest of the rules are the same as for Credit grades and will be discussed there. This result indicates that there is some quantifyable difference in the way these different populations of students perform in our examination even when all in session assessment results are the same.

The rules generated for a Credit using characteristic patterns were:

Characteristic pattern	Rule Set
ON Credit	(Crse ≤ 0.35) AND (Stage ≤ 0.93) AND (lab2 ≥ 0.39) AND (Midterm ≥ 0.12) AND (lab10 ≥ 0.14)

No other rule sets appear, because no single change has any significant effect. Multiple changes would be increasingly expensive to determine, and we can more easily reach the same end by increasing the discrimination of our characteristic patterns as mentioned previously.

The stage input indicates that surprisingly the few students who take this course quite late in their degree do not tend to achieve Credits! The appearance of lab2 and lab10 suggests that the material in those laboratory exercises were particularly relevant to good performance in the final examination.

Example of Explanation Facility Results

Using the previously described method rules and characteristic patterns for each of the outputs have been derived. An example of the explanations for this network is shown.

Student	Inputs													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
p0185591	0	0.5	1	0.8	0.7	0.7	0.7	1	0.8	0.4	1	0.3	0.49	0.5

p0185591
Network Output : Credit
Most Similar Characteristic Input : Credit
Important Inputs [Characteristic Values]
Crse [0.0] Stage [0.5]
Input 5 [0.85] Midterm [0.59]
Input 14 [0.5]
Satisfied Rule Set
(Crse ≤ 0.35) AND (Stage ≤ 0.93) AND
(lab2 ≥ 0.39) AND (Midterm ≥ 0.12) AND
(lab10 ≥ 0.14)
Next Most Likely Output : Distinction

It is interesting to note that in this case the student actually failed the mid session exam (the actual mark range was scaled to *0* to *1*), however the next most likely output predicted is a Distinction. This seems to be unusual, however the final grade was 71, so it is likely to be correct. The student likely got a high mark in the final exam. Nevertheless, the mark was predicted by the network and explained even though the result was not obvious.

(Crse is Input 1, Stage is Input 2, lab2 is Input 5, Midterm is Input 13, and lab10 is Input 14.)

Conclusion

Our explanation method reproduces the correct output in 94% of cases. Note that an explanation is correct if it matches the network's conclusion rather than if the conclusion is correct. We produce explanations in the form of rules, which may be useful for expert system knowledge acquisition [12]. Our method does not depend on the size or architecture of the network or on any unusual construction algorithm, but is general in nature. We have also demonstrated that useful educational knowledge can be extracted from a neural network trained on student marks to predict grades.

References

[1] Gedeon, TD and Bowden, TG "Heuristic pattern reduction," *Proc. International Joint Conference on Neural Networks*, Beijing, pp. 449-453, November 1992.

[2] Gedeon, TD and Bowden, TG "Heuristic Pattern Reduction II," *Proc. International Conference on Computer Science*, Invited Position Paper, Beijing, 1993.

[3] Sietsma, J & Dow, RF, "Creating Artificial Neural Networks That Generalize," *Neural Networks*, vol. 4, pp. 67-79, 1991.

[4] Gedeon, TD and Harris, D, "Network Reduction Techniques," *Proc. International Conference on Neural Networks Methodologies and Applications*, San Diego, vol. 2, pp. 25-34, 1991a.

[5] Gedeon, TD and Harris, D, "Creating Robust Networks," *Proc. International Joint Conference on Neural Networks*, Singapore, vol. 3, pp. 2553-2557, 1991b.

[6] Bochereau, L and Bourgine, P "Expert systems made with neural networks," *International Joint Conference on Neural Networks*, vol. 2, pp. 579-582, January 1990.

[7] Gallant, SI "Connectionist expert systems," *Communications of the ACM,* vol. 31, no. 2, pp. 152-169, February 1988.

[8] Klimasauskas, CC "Neural networks tell why," *DD Jour.*, April 1991.

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

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

[11] Turner, H and Gedeon, TD "Extracting Meaning from Neural Networks," *Proceedings 13th Int. Conf. on AI*, Avignon, 1993.

[12] Sestito, S and Dillon, T "The use of sub-symbolic methods for the automation of knowledge acquisition for expert systems," *Proc. 11th International Conference on Artificial Intelligence*, Avignon, 1991.