Extracting Meaning from Neural Networks

Harry S. Turner and **Tamás D. Gedeon**[1]

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

Ph: +61 2 697 4034
Fax: +61 2 313 7987

Abstract

Neural networks can be trained to provide solutions where clear rules do not exist which would allow symbolic solutions. Neural networks suffer from the disadvantage that there is no explanation for why a particular decision was made by the network.

We present our method of providing explanations. By using the causal index on charateristic input patterns, we produce a list 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.

Keywords explanation mechanisms; rule extraction; knowledge acquisition; machine learning; neural networks

1To whom all correspondence should be addressed.

1 Introduction

Artificial Neural Networks possess the ability to learn from examples making them a useful and powerful tool. One application that is becoming increasingly popular is that of using the ANN in the role of a human expert. The ANN has the advantage of reducing large amounts of expensive expert time by learning from example data being fed to it. This method has obvious advantages, however unlike conventional expert systems the ANN has no ability to provide any sort of explanation as to how it comes to its conclusions. Rather than storing sets of rules the knowledge contained in an ANN is stored in the weight values distributed throughout the entire network.

We have produced an explanation facility for Artificial Neural Networks that can be implemented on any three layered Neural Network. Explanations do not simply consist of a set of rules as to why the network came to its conclusion, they include identification of important factors in the input, and the next most likely output of the network.

2 Proposed Solutions

One solution attempts to couple neural networks with small rule based Expert systems [6-8]. The network makes decisions and the rule base explains them. It is even possible to take the inputs to the network and its output and 'confabulate' an explanation using a small rule based system. These methods are case specific, and reasonable results have been produced, but they do not satisfy our goal of a general explanation technique.

Gallant [9] presented a method of extracting the knowledge contained in the network and forming an explanation based on the fact that the neuron activation values have already been inferred by the network. A minimal subset of the currently known information that is sufficient to infer this value is then found, and an IF-THEN rule is generated. In principle by using this method it is possible to examine the entire network to provide an accurate explanation of the network's conclusions. However, even for a single neuron, the number of IF-THEN rules can grow exponentially with the number of inputs. As a result this method only produces satisfactory results if the network is very small.

Towell and Shavlik propose two methods of rule generation, both based on the assumption that the network has been constructed from some rule set using an algorithm such as KBAAN [4]. These are reported to closely reproduce (and can even exceed) the accuracy of the network. This method is called the N of M method and is designed to produce rules of the following form: *IF (N of following M antecedents are true) THEN ...*

Logical rule derivation [5] involves examining the internal structure of the ANN to produce logical rules. Firstly the Neurons are classified into 2 distinct classes: smooth and strong. A smooth neuron is a neuron that has more activation values in the range [f(-2),f(2)] than is does outside this range. A neuron is strong if it is not smooth. The network is trained to cause the neurons to behave as either very smooth or very strong. As a result strong neurons can be interpreted as Boolean variables without effecting the performance of the network. Input neurons are also considered as Boolean, hence they are interpreted as strong neurons. To allow better semantic interpretations of neurons the inputs to each neuron are clustered into k classes depending on their weight. Each class is given an average weight coefficient w_k. A new intermediate numerical variable is introduced to ease further interpretation. Then considering the output or hidden neurons, a logical rule for an assumed activation level s can be extracted:

Assuming most of the x_{jk} are from strong neurons and thus Boolean, it is easy by enumeration to extract a disjunctive normal form. This is done by superposing all the possible cases with the *OR* connector, producing an equivalent equation at activation level s. Explicit enumeration is affordable as the number of variables has been greatly reduced by the introduction of intermediate variables.

Sensitivity Analysis [2] 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*.

Another method, also aimed at finding the effect that an input neuron *i* has on an output neuron *k* uses the fact that the networks activation functions are differentiable. Instead of simply changing input values by small amounts, this method uses mathematical means to find the rate of change of *k* with respect to *i* [1, 3]. This is the Causal Index (C_{ki}), and indicates the relationship between the k^{th} output and the i^{th} input neurons. 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 can be given by using equations such as: If C_{ki} is Positive and Large then 'If *i* is large then *k* is large'.

To produce a general solution to the task of justifying and explaining conclusions made by ANNs, the Causal Index relationship of inputs to outputs was examined. The problem was then to interpret it to produce accurate, understand-able explanations which describe key factors and their relationships.

3 Causal Index

Consider the three layer network described in Figure 1. The outputs of the neurons in the network are given by the formulae:

$$y_k = f(U_{k2}) = (1 + e^{-Uk2})^{-1} \qquad h_j = f(U_{j1}) = (1 + e^{-Uj1})^{-1}$$

$$U_{k2} = \sum_j w_{jk}\, h_j \qquad U_{j1} = \sum_i w_{ij}\, x_i$$

Where
- w_{ij} : Weight from neuron *i* to neuron *j*.
- U_{in} : Sum *i* inputs in n^{th} layer.
- y_k : Output neuron.
- *f* : Sigmoid function.
- h_j : Hidden neuron.

The rate of change of an output neuron y_k with respect to an input neuron x_i is found by calculating the derivative dy_k/dx_i using the chain rule of differentiation [1].

$$\frac{dy_k}{dx_i} = \frac{dy_k}{dU_{k2}} \cdot \frac{dU_{k2}}{dh_j} \cdot \frac{dh_j}{dU_{j1}} \cdot \frac{dU_{j1}}{dx_i} = f'(U_{k2}).f'(U_{j1}). \sum_j w_{jk}.w_{ij}$$

Yoda et al. then assume that the product $f'(U_{k2}).f'(U_{j1})$ is constant for all *k* and *j* [1]. The influence of x_i on y_k is thus statically determined by the weights. To verify this, the Causal Index is shown in Figure 2 using the full formula on an example network, holding three inputs constant and varying the fourth from *0* to *1*.

From this it can be seen that between *0* and *1* the value of the function varies. The Causal Index must be calculated with the full formula. As we can not use a static analysis of the weights as Yoda et al. do, a new way of interpreting Causal Indeces must be found. We have examined networks in which the func-tionality is known, generalised to other networks and tested for accuracy.

4 Characteristic Input

The complete formula used in calculating the Causal Index enhances the accuracy of results, however causes one major problem: the results are input specific. In effect, this means that the analysis can not be generalised to satisfy any set of possible inputs or even any arbitrary subset of the set of possible inputs. This problem has been 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 the input patterns will be broken into classes according to their effect on an output. When the input pattern causes the output being analysed to be turned on, it will be classed as an *ON* input pattern, otherwise it will be classed as an *OFF* input pattern.

Using the mean or median of 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.

5 Deriving Logical and Numeric formulae from the network

For the interpretation of Causal Index values, several networks trained to perform different logical functions were examined. Figure 3 shows the Causal Index for each of the four inputs in the characteristic *ON* input pattern for a network trained on the logical conjunction of the four inputs. The four separate curves in this graph are almost identical. In each case, the network's result is completely dependent on the presence of that input. Inputs producing curves of this form, that completely change the output (eg, from off to on) are interpreted as crucial inputs, which are combined using the conjunction operator.

This interpretation is found using the characteristic *ON* input pattern. Similar analysis of the graph of the Causal Index for the characteristic *OFF* input pattern for the network trained to produce the logical disjunction of four inputs was done. This also produced four almost identical graphs, but because they appear for the characteristic *OFF* input

pattern they are thus not crucial. Altering any of the four inputs substantially causes the output to become *ON*. Thus, such curves in characteristic *OFF* input patterns are combined by the disjunctive operator.

The analysis was extended to cover formulae such as conjunction of disjunctions and vice versa. The results of one of these is shown.

To examine the conjunction of dis-junctions a network was trained on the logical formula *((A OR B) AND (C OR D)*). The characteris-tic *OFF* input for the network yields the graph of its Causal Index shown in Fig-ure 4. It should be noted that the peaks in the rate of change of the output in this case do not indicate a complete change of network output (found by examining the networks output over the same change in input). Change in any single input in this pattern does not turn the output neuron *ON*. As a result, change in more than one of the inputs is required to produce an *ON* output. The inputs *A* and *B* are represented by the smaller two curves (appearing identical), and *C* and *D* by the larger two curves (also appearing identical). Using the previously devised interpretation Figure 4 indicates that turning any two of these inputs on at the same time will cause the output to change from off to on. The logical formula that would be produced under this interpretation from the graph is: *((A OR B) AND (C OR D)) OR ((A OR C) AND (B OR D)) OR ((A OR D) AND (B OR C)))*. This logical formula would allow the output to be ON in more cases than selected by the network, hence some limitation must be found.

The sum of the maximum rate of change of Inputs *A* and *C* and a similar sum of Inputs *A* and *D* are identical, likewise the sums of Input *B* and *C* and Inputs *B* and *D*. These are the only combinations that result in the sum of the maximum Rate of Change (Causal Index Value) being the same. This fact is desired as it is known that all inputs are equally important and therefore, should have the same effect (hence rate of change) on the output. For this reason the interpretation of such a graph is that multiple inputs not turning the output *ON*, but appearing as peaks in a graph of a characteristic *OFF* pattern are combined in pairs of equal (or very similar) maximum rates of change using the *OR* connector. These pairs are then combined using the *AND* connector.

From the above sections on logical formulae, the following interpretations are drawn.

- Inputs with a large peak in rate of change of an output (enough to turn the output on) in a characteristic *ON* pattern are combined using the conjunction operator.

- Inputs with a large peak in rate of change of an output (enough to turn the output on) in a characteristic *OFF* pattern are combined using the disjunction operator.

- Two inputs that appear on the graph of a characteristic *ON* pattern that do not turn the output completely off if set to zero are interpreted as their disjunction.

- Two inputs that appear on the graph of a characteristic *OFF* pattern that do not turn the output completely on when set to one, are interpreted as their conjunction.

- As a further restriction in the above two interpretations, if more than two inputs occur that do not completely turn an output on or off, the combinations are done by firstly combining inputs with the most similar graphs with the *AND* or *OR* connectors respectively, then these grouped inputs are combined by the opposite connector.

To extend the above formalisation to non-boolean inputs, a three layered neural network consisting of two inputs, two hidden units and one output unit was trained to recognise the following logical formulae: *((A < 0.2) OR (B > 0.6))*. The causal index curves of the Characteristic *OFF* input is shown in Figure 5.

The graph shows a large negative peak in the rate of change for Input *A* which then returns to a constant level (zero) at *0.2*, and a large positive peak in the rate of change for Input B which

returns to a constant level (zero) at *0.6*. This leads to the conclusion that the point at which the Rate of Change returns to a constant level is the value the input must reach. Being in a characteristic *OFF* input the logic is *((input 1 < 0.2) OR (input 2 > 0.6))* for the output to be on. This is the desired (and expected) conclusion.

This analysis leads to the conclusion that when dealing with real input values a cut off point is the position at which the rate of change begins to stabilise (usually when it returns to zero).This cut off point is then used for non-boolean interpretation (using the formalised interpretation previously given) of the input's relationship with the output.

It should be noted that the logical formulae described above is complete, all logical expressions can be expressed in these terms. For example *A XOR B* can be expressed in the form *(A AND ~B) OR (~A AND B)*, which is a disjunction of two conjunctions.

6 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 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.

7 Example of Explanation Facility Results

The network to be analysed is a three layer neural network with fourteen inputs, five hidden units and four output units. The network has been trained on a set of partial marks in a subject to predict the final mark that a student will receive [10-11].The classification of the marks are:

- Fail, represented by output 4.
- Pass, represented by output 3.
- Credit, represented by output 2.
- Distinction or above, represented by output 1.

Using the previously described method rules and characteristic patterns for each of the outputs have been derived. Examples of the explanations for this network are below.

Example 1

p0177212
Network Output : Pass
Most Similar Characteristic Input : Pass
Important Inputs [Characteristic Values]
Input 13 [0.39]
Satisfied Rule Set
(Input 13 < 0.75)
Next Most Likely Output : Credit

In this case the student ended up attaining a final result of 61%. The network output is therefore correct and the prediction of the next most likely output is consistent with the next nearest possible grade in value (the final mark for a Credit is 65%).

Example 2

p0194588
Network Output : Distinction
Most Similar Characteristic Input : Distinction
Important Inputs [Characteristic Values]
Input 13 [0.72] Input 7 [1.0]
Input 11 [1.0] Input 5 [0.7]
Satisfied Rule Set
(Input 5 > 0.38) AND (Input 7 > 0.23) AND
(Input 11 > 0.27) AND (Input 13 > 0.37)
Next Most Likely Output : Credit

This is a straightforward result, the most similar characteristic input pattern being that of the Distinction and the next most likely output being the Credit. The Input 13 is the mid-session quiz, and an input value of 0.72 is a good mark, the student performed well on key assignments, thus the result is consistent with experience.

Example 3

p0185591
Network Output : Credit
Most Similar Characteristic Input : Credit
Important Inputs [Characteristic Values]
Input 1 [0.0] Input 2 [0.5]
Input 5 [0.85] Input 13 [0.59]
Input 14 [0.5]
Satisfied Rule Set
(Input 1 < 0.35) AND (Input 2 < 0.93) AND
(Input 5 > 0.39) AND (Input 13 > 0.12) AND
(Input 14 > 0.14)
Next Most Likely Output : Distinction

It is interesting to note that in this case the student actually failed the mid session exam, 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.

Example 4

p0276349
Network Output : Fail
Most Similar Characteristic Input : Fail
Important Inputs [Characteristic Values]
Input 9 [0.0] Input 13 [0.21]
Satisfied Rule Set
(Input 9 < 0.52) AND (Input 13 < 0.98)
Next Most Likely Output : Pass

Again this is quite a straightforward result: the student obtained nearly a complete set of zeros, hence the most similar characteristic pattern is that of the fail (the characteristic pattern with the lowest values) and the next most likely ouptut is the pass.

8 Summary and Conclusion

Our method of extracting meaning from neural networks to provide a useful explanation facility takes the following format:

- Characteristic inputs for each output pattern are calculated, each of these input patterns is then used as both characteristic ON and OFF inputs for the separate outputs of the network.

- The Causal relationship of the characteristic input with respect to the outputs in the network is calculated.

- This relationship is then used to determine which inputs are of importance to the outputs of the network.

- Rule sets using these important inputs are then generated.

- The input pattern is likened to the characteristic inputs, the characteristic input pattern showing the most similarity is selected and the important inputs and the satisfied rule set are presented as the explanation.

- The next most likely output is calculated and presented.

This explanation method only fails to produce the correct output in 6% of cases for the training set of the network used. The separate specification of results for training versus test cases is not required since the measure of success is the replication of the network's result. That is, an explanation is correct if it matches the conclusion rather than if the conclusion is correct.

By using our explanation method, many facets of the network's calculations can be determined. Firstly it becomes apparent which inputs are considered by the network for each output. The inputs considered for the *Distinction* class for example, are inputs *5*, *7*, *11* and *13*. To test the accuracy of this, the set of all inputs classified as *Distinction*s in the training set had all other inputs set to zero. The resulting input patterns were input to the network, and in each case the *Distinction* classification was still chosen by the network. To examine the overall accuracy of the complete set of inputs considered important, all inputs not appearing in any of the set of inputs considered important to any of the outputs were set to zero in each pattern in the training set. The networks accuracy on the training set went down by 10% to 86%. Considering that over 40% of the input data has been discarded, this is a very good result.

Our explanation facility provides explanations in the form of rules, which may be useful for expert system knowledge acquisition [12-13].

The rule set derived using this method is limited in application only by the selection of which rule is to be used. This selection is currently performed by likening the input pattern with the characteristic patterns.

The presentation of the next most likely output by the explanation facility is in this case only useful for the sake of interest of the user. Other applications of the next most likely output may include:

- Providing a 'safety net' in the case of incorrect classification, and

- Providing soft limiting boundaries – making decisions in 'grey areas' (such as classifying a mark of 64% as a Pass or a Credit) more flexible.

The accuracy of our method is affected by the size of the network's training set. *Characteristic* inputs will be most accurate using large training sets, which will allow statistical methods more accurate than simply finding the mean or median to generate them. Nevertheless, we have achieved good results with a training set of only 84 patterns. Our method does not depend on the size or architecture of the network or on any unusual construction algorithm, but is general in nature.

References

[1] 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.

[2] Klimasauskas, CC "Neural networks tell why," *Dr. Dobbs Journal*, April 1991.

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

[4] Towell, GG and Shavlik, JW "The extraction of refined rules from knowledge-based neural networks," *Machine Learning*, August 1991.

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

[6] Caudill, M "Making an expert network," *AI - Expert*, pp. 41-45, July 1990.

[7] Caudill, M "Using neural networks," *AI - Expert*, pp. 49-54, November 1990.

[8] Caudill, M "Expert networks," *Byte*, pp. 108-116, October 1991.

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

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

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

[13] Sestito, S and Dillon, T "Automated knowledge acquisition of rules with continuously valued attributes," *Proc. 12th International Conference on Artificial Intelligence*, Avignon, 1992.

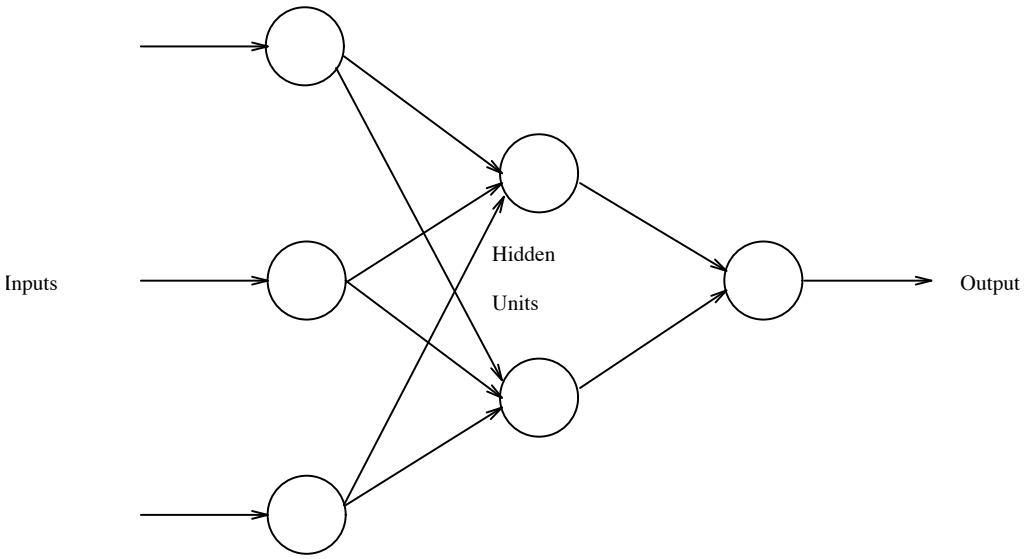

Figure 1 Structure of a three layered ANN with one output

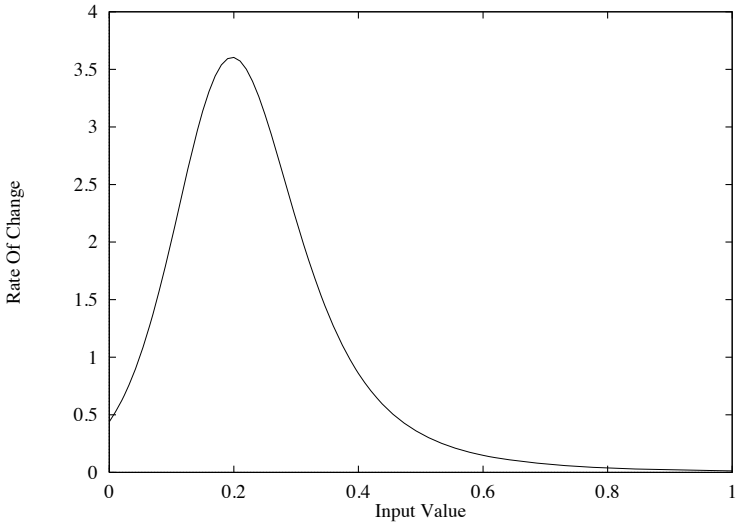

Figure 2 All inputs initially OFF

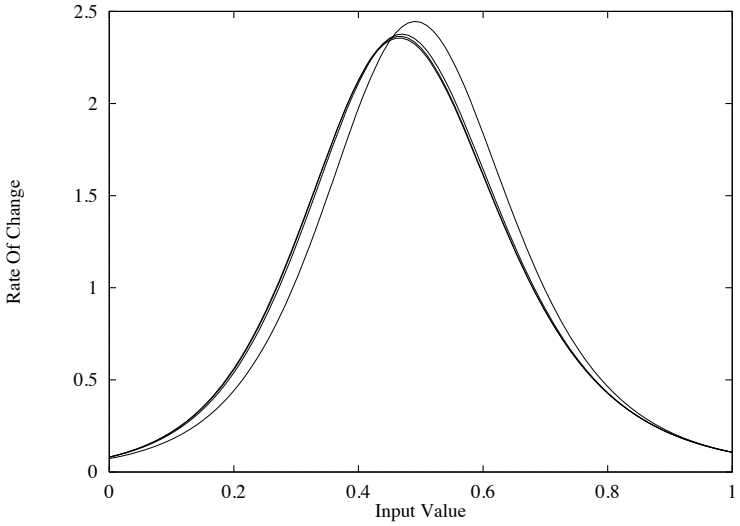

Figure 3 Causal Index of each input in the AND network

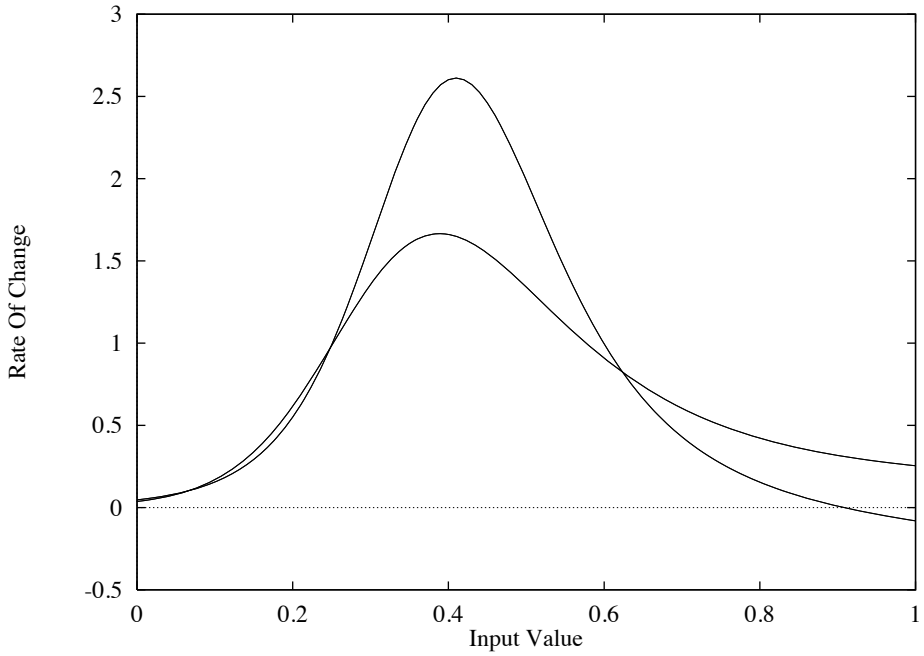

Figure 4 Causal Analysis of the Conjunction of Disjunctions network

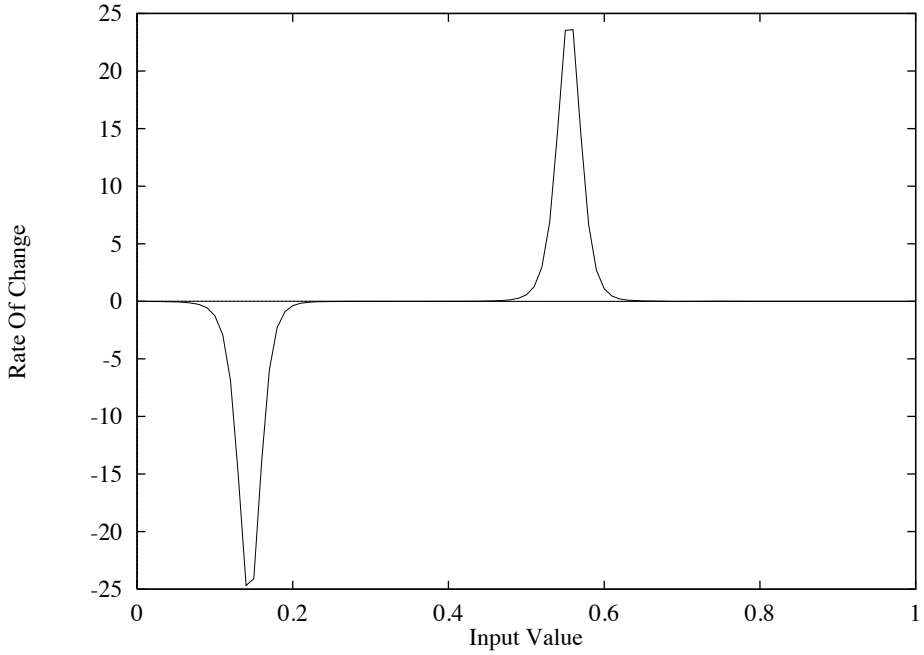

Figure 5 Causal Index of Non Boolean OR network