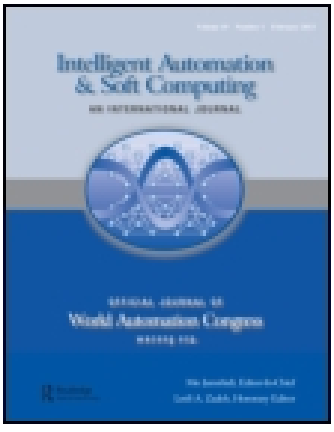

Intelligent Automation & Soft Computing
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tasj20

Abstracting Uncertain Knowledge: Case for Neural Nets Application

Tamás Gedeon[a], Arthur Ramer[a], Colette Padet[b] & Jacques Padet[b]
[a] School of CSE, University of New South Wales Sydney 2052, Australia
[b] Laboratoire de Thermomécanique, S.U.E., GRSM B.P. 347, 51062 Reims Cedex, France
Published online: 24 Oct 2013.

To cite this article: Tamás Gedeon, Arthur Ramer, Colette Padet & Jacques Padet (1995) Abstracting Uncertain Knowledge: Case for Neural Nets Application, Intelligent Automation & Soft Computing, 1:4, 365-377, DOI: 10.1080/10798587.1995.10750642

To link to this article: http://dx.doi.org/10.1080/10798587.1995.10750642

Intelligent Automation and Soft Computing, Vol. 1, No. 4, pp. 365-378, 1995

ABSTRACTING UNCERTAIN KNOWLEDGE: CASE FOR NEURAL NETS APPLICATION

TAMÁS GEDEON AND ARTHUR RAMER
School of CSE, University of New South Wales
Sydney 2052, Australia

COLETTE PADET AND JACQUES PADET
Laboratoire de Thermomécanique, S.U.E., GRSM
B.P. 347, 51062 Reims Cedex, France

ABSTRACT—Neural networks have been used successfully in practical applications where human expertise exists but no clear rules are known. There is a more difficult case when the acquisition of labelled data points is very expensive, such as in labelling of ground data to match satellite images in geographic information systems.

In fact, the dependence of neural networks on large volumes of training data result in the neural solution, producing more inconsistent results over a number of trials using the same data, but different initialisations of the weights.

We present our method of generating IF-THEN rules expressing the trained neural network's behaviour. By using the *causal index* on characteristic input patterns, we produce a list of inputs which were significant in reaching the decision made and a well-ordered sequence of rules governing this decision. This method correctly produced rules for 94% of the decisions made by a sample network.

The principle of selecting the next most likely decision (that the network could have made) brings forth the question of specificity of the ensuing procedure. The extent to which each rule "outweighs" its successor implies the degree of our confidence in the correctness of the final decision. Viewing causal indices as degrees of *possibility* of rules relevance permits using the machinery of formal *specificity measures* to capture this notion quantitatively. We outline the methodology, which has a well-defined axiomatic basis, and leads to a parametrized family of specificity functions.

Key Words: rule extraction, causal index, neural networks, specificity measures, possibility theory

1. INTRODUCTION

The network used is a three layered neural error backpropagation[1] neural network consisting of fourteen inputs, five hidden units and four output units. The network has been trained on a set of assessment marks in an undergraduate Computer Science subject to predict the final result that a student will receive.[2,3] The assessments shown are combined to yield 40% of the total mark, the last 60% being the exam which is omitted from the neural network training. Thus, the task is to predict the final mark which was calculated using the exam mark, from only the 40% of class assessments during the teaching session. The marks fall into the following categories:

- Distinction or above, being a mark of ≥ 75,
- Credit, being a mark between 65 and 74,
- Pass, being a mark between 50 and 64, and
- Fail, being a mark less than 50.

We augment the networks with a specially designed explanation facility[2] for back-propagation trained neural networks. 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. This method bring our system output on par with conventional expert systems, in that it has the ability to provide rule traces and some sort of explanation as to how it comes to its conclusions.[3]

Lastly we discuss the conceptual and numerical framework, based on possibility theory, for assigning specificity values to conclusions.

2. RELATED WORK

The number of rules that can be extracted from a trained neural network is potentially exponential in the number of inputs used. We present a taxonomy of the different rule extraction techniques based on how the problem of combinatorial explosion is avoided. There are four components: (i) how is the search space for rules reduced, (ii) is the learning algorithm modified, (iii) how is the search carried out, and (iv) is the network treated as a black-box.

The search space for rules can be reduced by clustering. The weights can be clustered,[4-6] or the hidden units,[7] or the input patterns. In this paper we cluster input patterns. The backprogation algorithm is modified in most techniques[5,6] which use weight or hidden unit clustering. These modifications are to add extra cost functions to the training. If the input patterns are clustered, modifications of learning do not provide obvious advantages, allowing us the advantage of maintaining the standard form of the algorithm and hence wider applicability. The search can be exhaustive,[4] but is generally carried out by using heuristic optimisations.

It is clear from the literature that rule extraction from with a black-box approach is not satisfactory. Approaches which use only the weight matrix[7-9] have been successful, though with problems as described in the next section.

3. CAUSAL INDEX

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

For a three layer network, the outputs of the neurons in the network are given by the formulae:

$$y_k = f(U_{k2}) = \left(1 + e^{-U_{k2}}\right)^{-1} \qquad U_{k2} = \sum_j w_{jk} h_j$$

$$h_j = f\left(U_{j1}\right) = \left(1 + e^{-U_{j1}}\right)^{-1} \qquad U_{j1} = \sum_i w_{ij} x_i$$

The rate of change of an output neuron yk with respect to an input neuron xi is found by calculating the derivative dy_k / dx_i using the chain rule of differentiation.

$$\frac{dy_k}{dx_i} = \frac{dy_k}{dU_{k2}} \cdot \frac{dU_{k2}}{dh_j} \cdot \frac{dh_j}{dU_{j1}} \cdot \frac{dU_{j1}}{dx_i}$$

$$= f\left(U_{k2}\right) \cdot f\left(U_{j1}\right) \cdot \sum_j w_{jk} \cdot w_{ij} = C_{ki}$$

This method has also been used in,[4,5] using the assumption that the product $f'(U_{k2}) \cdot f'(U_{j1})$ is constant

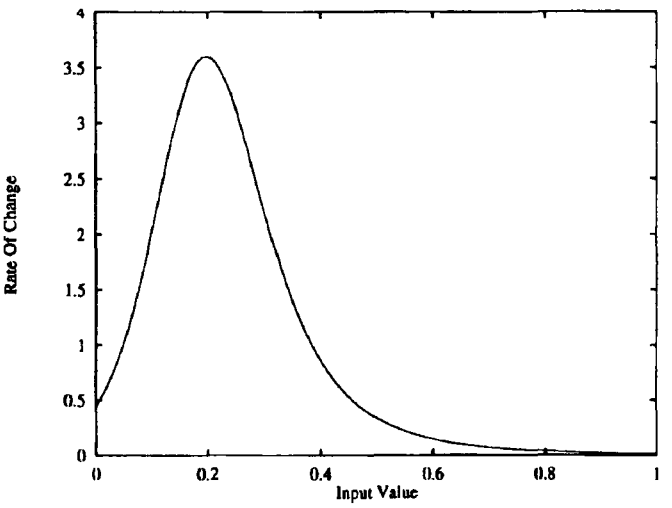

Figure 1. All inputs initially OFF.

for all k and j. The influence of x_i on y_k could thus statically be determined from the weight matrix of the trained network.

While this assumption may hold in some domains, we have found that this does not hold in the domains we have tried. Figure 1 demonstrates the value of the product $f'(U_{k2}) \cdot f'(U_{j1})$ in a simple four input, two hidden unit, and one output network, by holding three of the inputs constant, and varying the fourth from 0 to 1. As we can readily see, the value varies, and is not close to a constant.

To produce a general solution to the task of justifying and explaining conclusions by the Causal Index, relationship of inputs to outputs needs to be interpreted to produce accurate and understandable explanations which describe key factors and their relationships. This interpretation of the Causal Index is done by using the *Characteristic patterns*. We have examined simple networks in which the functionality is known, then generalised to other networks and tested for accuracy. This article includes an extended example using real data.

4. CHARACTERISTIC PATTERNS

The formula used in calculating the Causal Index causes one major problem: the results are input specific. 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 not very efficient, and 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 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 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 the mean or median of the values of each input variable, a pattern representing each input class is created as a *Characteristic* pattern for that class. A characteristic *ON* pattern represents those input patterns which turn an output on. Similarly a characteristic *OFF* pattern represents those input patterns which turn an output off. We had expected to need to use more statistically mature means of finding the *Characteristic* patterns for inputs, such as clustering techniques to find a number of patterns for each output class, however in this example this has not been necessary. Note that we consider here networks where the output units represent category membership by high output values. For networks where an output unit's value is used directly and not just as a threshold, we can define *Characteristic* patterns over subranges of the value of the output unit activation before clustering.

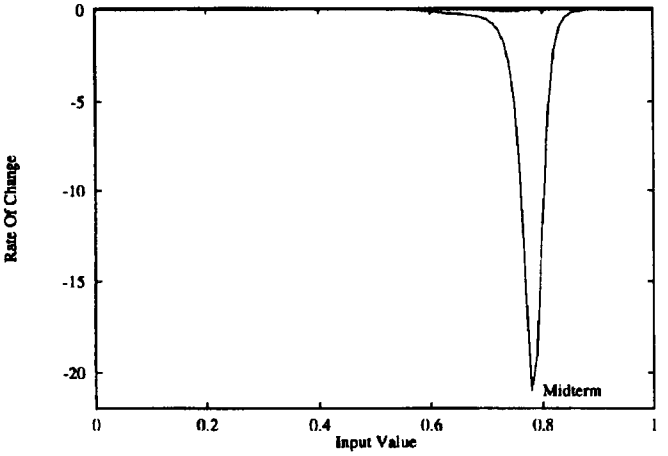

Figure 2. Casual Index of Inputs for Mark Predictor network—CON_{Pass}.

Table I
Rules Produced from the Causal Index Graph for Mark Predictor Network—CON_{Pass}

Characteristic Pattern	Rule Set
CON_{Pass}	Midterm < 0.75

TABLE II
***Characteristic* ON Pattern for the Pass Output—CON_{Pass}**

Course	Stg	Enr	Tutgp	lab2	TutAss	lab4
0.65	0.85	0.98	0.67	0.52	0.44	0.52
H1	H2	lab7	P1	F1	Midterm	lab10
0.67	0.65	0.5	0.52	0.42	0.39	0.49

TABLE III
***Characteristic* OFF Pattern for the *Credit* Output—$COFF_{Pass}^{Dist}$**

Course	Stg	Enr	Tutgp	lab2	TutAss	lab4
0.7	1.0	1.0	0.5	0.7	0.4	1.0
H1	H2	lab7	P1	F1	Midterm	lab10
0.8	0.8	0.7	1.0	0.7	0.72	0.5

Note: P1 is Procedural assignment 1.

5. PASS OUTPUT

The output context for the rules derived in this section will be the Pass decision made by the network, and we will derive these rules using the *Characteristic* pattern contexts.

5.1 *Pass Result with Input Pattern Similar to CON_{Pass}*

This is the simplest case, and we would expect to handle the largest number of examples using these rules. This is of course because the network decision matches with the most similar *Characteristic* pattern, and hence the explanation is in terms of the majority population. The situation calls for rules expressing why the actual pattern has been categorised (by the network) to belong to the category it seems to belong to (by similarity). The Causal Index graph is shown in Figure 2. The graph indicates that a single rule can be extracted. This is shown in Table I.

The single rule in Table I seems all encompassing, taken out of context. In the context of an explanation for patterns which are most similar to the *Characteristic ON* pattern for the *Pass* output ($CON_{Pass)}$ shown in Table II, it makes sense. That is, from all of the patterns which are similar to the classic *Pass* pattern, those which obey the rule are *Pass* results. Those which do not obey the rule, and are similar to the classic *Pass* pattern must belong to some other class.

5.2 *Pass result with input pattern similar to CON_{Dist}*

This is an unfortunate situation from the student viewpoint, in that the profile is largely that of a *Distinction* for the continuous assessment parts, but the overall decision is

only a *Pass* grade.

The Causal Index graph using the *Characteristic ON* for the *Distinction* as the *Characteristic OFF* for the *Pass* ($COFF_{Pass}^{Dist}$) is shown in Figure 3.

The graph shows that three variables have a significant effect. In this case each of them is sufficient to modify the output value on their own and thus are simply combined using the *OR* operator, as this is a *Characteristic OFF* pattern. The most similar *Characteristic* pattern (for the *Distinction* result) is shown in Table III.

The rules produced using the Causal Index graph in Figure 3, to distinguish patterns from the *Characteristic* pattern they are most similar to, are shown below, in Table IV.

The rule sets are interpreted such that one or more of the conditions from the *COFF* replace the appropriate variable in the *CON*. This produces six possible rules from Table IV, where the form used is a shorthand to clarify presentation of the actual rules for the $COFF_{Pass}^{Dist}$.

From the rules, we can conclude that good overall results combined with very low results in *lab4* or *P1* produces a *Pass*. Similarly, a low *Midterm* mark (in the bottom third of the class) indicates that a *Pass* result is likely. This makes sense as 60% of overall marks come from the final examination, and is not included in training data of the network.

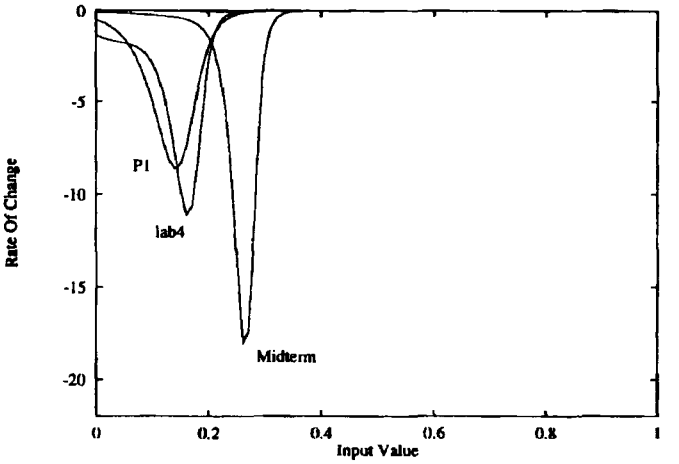

Figure 3. Casual Index of inputs for Mark Predictor network—$COFF_{Pass}^{Dist}$.

Table IV
Rules Produced from the Causal Index Graph for Mark Predictor Network—$COFF_{Pass}^{Dist}$

Characteristic Pattern	Rule Set
CON_{Dist}	(lab2 ≥ 0.44) AND (lab4 ≥ 0.23) AND (P1 ≥ 0.27) AND (Midterm ≥ 0.37)
$COFF_{Pass}^{Dist}$	(lab4 < 0.03) OR (P1 < 0.05) OR

5.3 Pass Result with Input Pattern Similar to CON_{Cred}

In this case, the most similar *Characteristic* pattern is the *Credit* result, though the most similar pattern to the student's pattern was for a *Pass* grade. This is less drastic a change than that discussed in the preceding section, therefore we might expect that there would be a number of ways a single change would convert a *Credit* to a *Pass*. The Causal Index graph is shown in Figure 4.

The graph shows that indeed a number of possible rules can be extracted. It is worth noting that the *Course* input appears as both positive and negative peaks. The course status encoding was made using a relatively unfortunate initial choice. We have retained this encoding for the demonstration purpose this graph (and rules extracted) affords.

The *Credit* result is the appropriate *Characteristic* pattern which provides the context for the rules derived, as shown in Table V.

The fifteen possible rules derived for this case are in Table VI. The two peaks for the *Course* input are individually too small to turn the output on, and are thus connected using *AND*. The other input peaks are

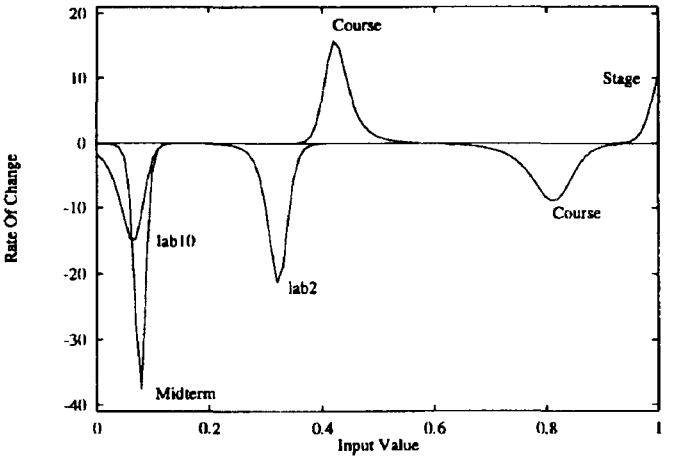

Figure 4. Casusal Index of Inputs for Mark Predictor network—$COFF_{Pass}^{Cred}$.

TABLE V
***Characteristic* OFF Pattern for the *Credit* Output—CON_{Pass}^{Cred}**

Course	Stg	Enr	Tutgp	lab2	TutAss	lab4
0.0	1.5	1.0	0.75	0.85	0.7	0.85
H1	H2	lab7	P1	F1	Midterm	lab10
0.8	0.8	0.55	0.7	0.3	0.59	0.5

Table VI
Rules Produced from the Causal Index Graph for Mark Predictor Network—$COFF_{Pass}^{Cred}$

Characteristic Pattern	Rule Set
CON_{Cred}	(Course < 0.35) AND (Stage < 0.93) AND (lab2 ≥ 0.39) AND (Midterm ≥ 0.12) AND (lab10 ≥ 0.14)
$COFF_{Pass}^{Cred}$	(Course ≥ 0.51) AND (Course < 0.63) OR (Stage ≥ 0.99) OR (lab2 < 0.22) OR (lab10 < 0.01) OR (Midterm < 0.02)

sufficiently large to produce separate rules.

Note that "large enough" and "too small" are adaptively thresholded, and depend on the actual effect on the relevant output unit.

The values of the *Course* input selected indicate that students from some of the non-science courses get *Pass* grades even when they otherwise perform to a *Credit* level by pattern similarity. The very high value for *Stage* denotes later year students exclusively. This is probably due to many later year students in this subject being repeat students who had failed. This is probably because term assessment is more likely to be similar to previous iterations of the course than the final exam. The very low marks on *lab10* and the *Midterm* exam are straightforward, as is the significance of the low mark for *lab2* indicating a *Pass* instead of a *Credit* result.

5.4 *Pass Result with Input Pattern Similar to CON_{Fail}*

Surprisingly, there is no single change which will produce a *Pass* result if the most similar pattern was the *Characteristic ON* pattern for the *Fail* output. From an educational viewpoint this is heartening, indicating the robustness of the *Fail* results, in that a single change in performance would not have boosted a *Fail* mark to a *Pass*.

5.5 *Example of Results*

The actual input pattern for a student is shown in Table VII.

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

This is a straightforward result, the most similar characteristic input pattern being that of the *Pass* and the next most likely output being the *Credit*. For the mid-session quiz (*Midterm*), an input value of 0.58 is a good mark, but not high enough to boost the mark to the next higher category. The student actually received 61% as the final mark, which is 4% short of a *Credit* result.

TABLE VII
Actual Input Pattern for a Student—p0177212

Course	Stg	Enr	Tutgp	lab2	TutAss	lab4
0.7	0.5	1.0	0.75	0.0	0.0	0.4
H1	**H2**	**lab7**	**P1**	**F1**	**Midterm**	**lab10**
0.2	0.6	0.4	0.3	0.0	0.58	0.0

6. RULE EXTRACTION SUMMARY

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 produced by clustering or numerically.
- The causal relationship of each of these characteristic inputs with respect to each of the outputs is calculated.
- This relationship is used to determine the inputs of importance to the outputs of the network.
- Rule sets using these inputs are then generated.
- The input pattern is compared to the characteristic inputs, the characteristic input pattern showing the most similarity is selected and the important inputs and the satisfied rule set is presented.
- The next most likely output is presented.

This method only fails to produce the correct rule 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, a rule is correct if it matches the conclusion of the network, 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 the inputs *lab2*, *lab4*, *P1* and *Midterm*. To test the accuracy of this, the set of all inputs classified as *Distinctions* 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 another 40% of the available 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.

The rule set derived using this method is limited in application only by the selection of which rule is

to be used. This 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 more accurate statistical methods to be used in their generation. 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.

In the following sections we discuss the numerical framework based on possibility theory, for assigning specificity values to conclusions.

7. DESIGN SPECIFICITY FUNCTIONS

We use as our model a countable infinite possibility distribution (p_i), $i = 1, 2, \ldots$.[10] Finite distributions will be usually identified with (p_i) where all but finitely many $p_i = 0$; in particular all $p_i \neq 0$ lie in an initial segment (p_i), $i = 1, \ldots, n$ for some n.

Our objective is to capture formally the intuition about specificity—a numerical value that might be assigned to any given distribution.[11,12] The main premise is the principle of *juxtaposition*:

> *Sp* (**p**) expresses the preference for a certain maximal p_0 over any and all the remaining p_i.

Now let us consider how, having selected $p_0 = \max(p)$, its *informal specificity* is estimated. We look first for the next largest p_i, a main competitor, and estimate how much it diminishes the specificity, compared to that of p_0 presented alone. The process is then iterated in the order of decreasing values of p_i, every next value considered causing the estimated specificity to diminish somewhat. We can picture it as a sequential process—an on-line algorithm given as input the decreasing rearrangement $(\tilde{p}_i)$.[13] We may also surmise that, for a given i, the drop in specificity caused by $\tilde{p}_i$ will not depend on the earlier inputs $\tilde{p}_i, \ldots, \tilde{p}_{i-1}$. This assumption of *independent influence* is devolved from the interpretation of *Sp* (**p**) as the specificity of its maximum, posed separately against of the remaining elements (*juxtaposition*).

Next, let us consider the effect of modifying the complete distribution (p) in a uniform manner. If we effect a multiplicative *scaling*, forming $\alpha p = (\alpha p_1, \ldots, \alpha p_n, \ldots)$, $0 \leq \alpha \leq 1$, we may assume that for any two distributions **p** and **q**, their relative specificities remain unchanged:

$$\frac{Sp(\alpha \mathbf{p})}{Sp(\alpha \mathbf{p})} = \frac{Sp(\mathbf{p})}{Sp(\mathbf{q})}$$

On the other hand, if the change consists of a uniform shift of values $\mathbf{p} - \beta = (p_1 - \beta, \ldots, p_n - \beta, \ldots)$ for $\beta \leq \inf_i p_i$, then no change of specificity should occur. This again results from *juxtaposition*, noting that for maximal p_0 and any given p_i, both the sequential placement and the difference in magnitude remain the same after the shift. We remark that such shift can be performed only for the infinite distributions.

The last item to be considered will be the effect of offering yet another choice, identical in value to several choices already provided. To make it more specific, let us consider a family of distributions $1^{(n)}$ (*n* values 1, and zeros other-wise). If $1^{(n-1)}$ is already given, and another value 1 'arrives', transforming the distribution into $1^{(n)}$, then the common perception of *specificity* is that the change due to such *n*-th choice will be ever less as *n* increases. This view is also strongly supported by the analysis presented in the next section. We demonstrate there that, in the numerical terms, the decrease of specificity due to the *n*-th value 1 has limit zero as *n* goes to infinity. The statement above makes that limit to be reached monotonically.

Although in absolute terms the influence of each additional 1 decreases, it does not say much about its *relative* effect. With rather less assurance, we can postulate that the arrival of each consecutive 1 takes away the same proportion of specificity $Sp\left(1^{(n-1)}\right)$ still available. After all, we consider yet another *identical* choice; only we consider it at stage *n* and not sooner.

In the next section we study the implications of the proposed properties. We do it first without the last assumption, and then use it to derive a particularly simple formula for computing specificity.

So far, we did not mandate any fixed values of the specificity for any special distributions. For definiteness we need to establish the range of the admissible values and associate its end-points with concrete distributions. We do it now by allowing [0,1] range of values and assigning the upper value to $\mathbf{p} = (1,0,\ldots)$. The least specific distributions will be of the form $(c,\ldots,c,\ldots)$, in particular $(1,\ldots,1,\ldots)$. Now the possibility values do not distinguish the elements at all and we put $Sp(1,\ldots,1,\ldots) = 0$.

8. FUZZY SPECIFICITY

We proceed to extract an analytical representation from the rules elaborated in the previous section.[14] Our model will consist of a *specificity* function *Sp* (**p**), taking values in [0,1] and defined on the space of infinite sequences $(p_1,\ldots,p_n,\ldots)$, $0 \le p_i \le 1$. We assume *specificity* to be invariant under the permutations of (*p*) and, more significantly, under the descending rearrangements of (*p*). First condition is trivial, as permutations of (*p*) simply correspond to the isomorphic fuzzy sets. Second condition we find both natural and permissible within the framework we trying to formalise.[15]

Informal specificity is estimated by selecting an element of the maximum value, then considering the alternatives, one by one in the descending order of values. For an infinite (*p*) certain values could be indefinitely delayed, thus allowing us to ignore those values altogether. This is exactly equivalent to replacing (*p*) by $(\tilde{p})$ and computing $Sp(\tilde{p})$. In the future we make this substitution automatic, implicitly assuming the distribution given as $p_1 \ge p_2 \ge \ldots$. Now the selected maximal element will become $\tilde{p}_1$.

From the assumption of *independent influence* we find

$$Sp(\tilde{p}_1,\ldots,\tilde{p}_n,\tilde{p}_{n+1}) - Sp(\tilde{p}_1,\ldots,\tilde{p}_n)$$

to be constant with respect to $\tilde{p}_1,\ldots,\tilde{p}_n$, therefore a function of $\tilde{p}_{n+1}$ only. Writing

$$f_n(x) = Sp(\tilde{p}_1,\ldots,\tilde{p}_n,x) - Sp(\tilde{p}_1,\ldots,\tilde{p}_n)$$

we have a sum form

$$Sp(\mathbf{p}) = \sum_{i=1}^{\infty} f_i(\tilde{p}_i)$$

The rule about scaling can be written as

$$\frac{Sp(\alpha \mathbf{p})}{Sp(\mathbf{p})} = g(\alpha)$$

Combining with the previous formula gives

$$f_i(x) = g(x) f_i(1)$$

and denoting

$$w_1 = f_1(1),\ w_1 = -f_i(1),\ i = 2,\ldots$$

we have

$$Sp(\mathbf{p}) = w_1 g(\tilde{p}_1) - \sum_{i \geq 2} w_i g(\tilde{p}_i)$$

we use this notation to stress the role of the maximum possibility value. However, scaling implies much more than the existence of function $g(\alpha)$—for arbitrary α, β

$$\frac{Sp(\alpha\beta \mathbf{p})}{Sp(\beta \mathbf{p})} = \frac{Sp(\alpha \mathbf{p})}{Sp(\mathbf{p})}$$

whereupon

$$g(\alpha\beta) = g(\alpha) g(\beta)$$

For a continuous $g(\alpha)$—necessary, because of the continuity of $Sp(\mathbf{p})$—it means $g(\alpha) = \alpha^k$, for some constant k and

$$Sp(\mathbf{p}) = w_1 \tilde{p}_1^k - \sum_{i \geq 2} w_i \tilde{p}_i^k$$

Before we determine constant k, we establish a few important properties of the coefficients w_i. Recalling that the specificity of $(1,0,\ldots)$ equals 1 and applying the last formula gives $w_1 = 0$. The requirement that $Sp(\mathbf{p}) > 0$ unless all $\tilde{p}_i$ are identical gives $w_i > 0$, $i = 2, \ldots$. Finally, computing $Sp(1, \ldots, 1, \ldots)$ yields $\sum_{i \geq 2} w_i = 1$.

We can compute $Sp\left(1, \frac{1}{2}, \ldots, \frac{1}{2}, \ldots\right)$ in two different ways. First

$$1 - \sum_{i \geq 2} w_i \frac{1}{2^k} = 1 - \frac{1}{2^k}$$

Next, by applying an additive shift we will not change the specificity. Shifting by $\frac{1}{2}$ gives $\left(\frac{1}{2}, 0, \ldots\right)$ and the specificity $\frac{1}{2^k}$. Equating those values yields $k = 1$ and a linear representation

$$Sp(\mathbf{p}) = \tilde{p}_1 - \sum_{i \geq 2} w_i \tilde{p}_i$$

with $\sum_{i \geq 2} w_i = 1$. Summation can be viewed as finite or infinite depending on the representation of

distribution **p**. From here we can conclude that $\lim_{i\to\infty} w_i = 0$, though they need not be monotonically decreasing. We establish this last property by appealing to the property of "diminishing returns" —each next equal value diminishes the specificity even less.

Let us apply it to the distributions of the form $1^{(n)}$; as $Sp\,(1^{(n-1)}) - Sp\,(1^{(n)})$ measures the effect of the n-th value 1, it decreases steadily. Its numerical value is w_n, giving $1 > w_2 > w_3 > \ldots$.

We consider the linear form of $Sp\,(\mathbf{p})$ with mono-tonically decreasing coefficients a *general* form of the specificity function. It is general enough to fit most applications and, if w_i are supplied, it offers a comparison scale among the distributions.

Coefficients w_i can be established much more precisely by appealing to the last property discussed in the previous section. Again looking at $1^{(n)}$, the relative constancy of the effect of each value 1 means

$$\frac{Sp\left(1^{(n-1)}\right)}{Sp\left(1^{(n)}\right)} = \text{const}$$

Therefore

$$\frac{1-\sum_{2}^{n} w_i}{1-\sum_{2}^{n-1} w_i} = \text{const}$$

for all n. An easy calculation produces

$$w_i = \omega^{i-1} - \omega^{i}$$

for some ω, $0 < w < 1$. We obtain a *definite* form of specificity

$$Sp(\mathbf{p}) = \tilde{p}_i - \sum_{i\geq 2}\left(\omega^{i-1} - \omega^{i}\right)\tilde{p}_i$$

In many applications selecting the actual value of ω is not critical; its role, as we shall see later, is similar to that of the base of logarithms. Choosing $\omega = \frac{1}{2}$, which has certain correspondence to binary logarithms, gives

$$Sp(\mathbf{p}) = \tilde{p}_i - \sum_{i\geq 2}\frac{1}{2^{i-1}}\tilde{p}_i$$

In the formulas we derived the role of $\tilde{p}_1$ is manifestly different from that of $\tilde{p}_i$, $i \geq 2$. A more symmetric expression can be obtained using Abel summation formula. Defining $W_i = 1 - w_2 - \ldots - w_i$ gives the *general expression*

$$Sp(\mathbf{p}) = \sum_{i\geq 1} W_i\left(\tilde{p}_i - \tilde{p}_{i+1}\right)$$

and the *definite one* $Sp(\mathbf{p}) = \sum_{i\geq 1}\omega^{i-1}\left(\tilde{p}_i - \tilde{p}_{i+1}\right)$.

They play a very important role in the development of the *continuous* model of specificity.[16]

9. CONTINUOUS SPECIFICITY MODEL

We propose a two-part structure, depending on the measure of the domain of discourse.

If $\mu(X)$ is finite we rearrange it to form $\tilde{f}(x)$ on $[0, \mu(x))$. Then we propose as the basic measure

$$I(f) = \int_0^{\mu(X)} \frac{1-\tilde{f}(x)}{x}\, dx$$

It is equivalent to the familiar integral over [0,1] if we perform *scaling*, replacing $\tilde{f}(x)$ with $\tilde{f}'(x) = \tilde{f}(\mu(X)x)$ defined over [0,1]. We have

$$I(f) = \int_0^{\mu(X)} \frac{1-\tilde{f}(x)}{x}\, dx$$

$$= \int_0^1 \frac{1-\tilde{f}(\mu(X)x)}{\mu(X)x}\, d(\mu(X)x)$$

$$= \int_0^1 \frac{1-\tilde{f}'(x)}{x}\, dx$$

For X of infinite measure we propose using

$$Sp(f) = k\int_0^\infty \tilde{f}(x)e^{-kx}\, dx$$

or, in general

$$Sp(f) = \int_0^\infty \tilde{f}(x)W'(x)\, dx$$

for $W(x)$ —a monotonically decreasing function satisfying

$$W(0) = 1 \,.\, \lim_{x\to\infty} W(x) = 0$$

10. RESULTS AND FUTURE WORK

We have used fifty patterns set aside as a test set and never used in training. The remaining 100 patterns were used to create 50 sets of 70 pattern training sets at random. Fifty networks were trained and the integrated bias and variance were then calculated.[17] The use of the specificity measures as possibility values as described allows inconsistent results over multiple training trials to be handled, as successive possibility values gives us increased confidence in the correctness of the decision, beneficial to overall system performance.

The consistency between the neural networks and the rules we produce are 94% on the training set, which is satisfactory. The generalisation from the rules is 75% correct predictions on the test set, which

is identical to the network performance also. Our rules are concise and readable. We have included in this paper all of the rules for the *Pass* student result, which comprise some 40% of the total number of rules found for this data set.

A direction we are investigating in our work is to apply our specificity measures to other neural network rule extraction methods based on different means of reducing the search space.[6,7]

11. CONCLUSION

We have presented our method for generating explanations from trained neural networks, and outlined the conceptual and numerical framework based on possibility theory, for assigning specificity values to conclusions which leads to a parametrized family of specificity functions.

This allows us to use the sometimes inconsistent results from neural networks over a number of (training) trials using the same data and use the machinery of formal *specificity measures* to capture this notion quantitatively.

ACKNOWLEDGEMENTS

We would like to thank the anonymous referees for helpful suggestions, particularly with regard to the suggestions for future work.

REFERENCES

1. Rumelhart, D.E., G.E. Hinton, R.J.Williams. "Learning internal representations by error propagation." D.E. Rumelhart, ed. *Parallel Distributed Processing*, vol. 1, MIT Press, 1986.
2. Gedeon, T.D., and H. Turner. "Explaining student grades predicted by a neural network." *Proc. Int. Joint Conf. on Neural Networks*, Nagoya, 1993, pp. 609-612.
3. Gedeon, T.D., and H. Turner. "Extracting Contextual if-then Rules from a Feedforward Neural Network." *Proceedings Brazil-Japan Joint Symposium on Fuzzy Systems*, 10 pages, Manaus, 1994.
4. Gallant, S.I. "Connectionist expert systems." *Communications of the ACM*, vol. 31, no. 2, February 1988, pp. 152-169.
5. Towell, G.G. and J.W. Shavlik. "The extraction of refined rules from knowledge-based neural networks." *Machine Learning*, August 1991.
6. Craven, M.W. and J.W. Shavlik. "Learning Symbolic Rules Using Neural Networks." *10th International Machine Learning Workshop*, Amherst, 1993, pp. 73-80.
7. Fu, L.M. "A Neural Network. Model for Learning Rule-Based Systems." *International Joint Conference on Neural Networks*, vol. 1, Baltimore, 1992, pp. 343-348.
8. Yoda, M., K. Baba, and I. Enbutu. "Explicit representation of knowledge acquired from plant historical data using neural networks." *International Joint Conference on Neural Networks*, San Diego, vol. 3, 1991, pp. 155-160.
9. Hora, N., I. Enbutsu, and K. Baba. "Fuzzy rule extraction from a multilayer neural net." *Proc. IEEE*, vol. 2, 1991, pp. 461-465.
10. Dubois, D., and H. Prade. *Possibility Theory*, Plenum Press, New York, 1988.
11. Ramer, A. and C. Padet. "Information semantics of possibilistic uncertainty." *Fuzzy Sets and Systems*, Special Issue (in print), 1994.
12. Yager, R. "On measuring specificity." Tech. Rep., Iona College, New Rochelle, NY, 1990.
13. Hardy, G., J. Littlewood, and G. Polya. *Inequalities*, Cambridge University Press, Cambridge, 1934.
14. Ramer, A., and R. Yager. "Specificity as uncertainty metric." 3rd IPMU - Int. Conf. Information Proc. and Management of Uncertainty, Madrid, 1992.
15. Dubois, D., and H. Prade. "Properties of measures of information in evidence and possibility theories." *Fuzzy Sets and Syst.* vol. 24, no. 2, 1987.
16. Ramer, A. "Concepts of fuzzy information measures on continuous domains." *Int. J. Gen. Syst.*, vol. 17, no. 2-3, 1990.
17. Slade, P., and T.D. Gedeon. "Bimodal Distribution Removal." *New Trends in Neural Computation*. J. Mira, J.Cabestany and A. Prieto (eds.) Springer Verlag. Lecture Notes in Computer Science, Vol. 686 (1993), pp. 249-254.