(isbn 2-909214-44-3)

Proceedings Intern. AMSE Conference "Systems", London (U.K.), Sept. 1-3, 1993
AMSE Press, Vol. 2, pp. 23-32

Shaping a Fuzzy Rule-Base: A Neural-Fuzzy System Which Learns Heuristically

Ahmet Şekercioğlu* Gregory K. Egan†

August 25, 1993

Abstract

In this study, an experimental self-learning neural-fuzzy system is presented. The system learns by doing and improves the performance. It consists of three main functional units: **Behavior Generation Unit**, which is a slightly modified fuzzy logic based controller; **Decision Unit**, which is a Bidirectional Associative Memory, and **Performance Evaluation Unit**. The learning and generalization capabilities of the system is demonstrated on the truck back-up problem.

1 Introduction

Humans have a demonstrated capacity to reach decisions under the limitations of imprecise, incomplete, ill-defined information; of course sometimes these decisions are wrong. Fuzzy mathematical techniques allow us to capture this type of reasoning and encapsulate it within artificial systems. Increasingly, controllers based on fuzzy reasoning are being applied to several commercial products successfully but, they lack one important ability of human beings: learning and generalisation. Learning from experience and an ability to generalise make us dynamic and adaptive. Experience, trial and error shape the mind and so, change our responses. On the other hand, in the fuzzy controllers, once expressed, the rule-base is usually fixed and static. To overcome this deficiency, the learning and generalisation capability of the artificial neural networks can be used in a combined system.

In this study, an experimental, unsupervised learning neural-fuzzy system is presented. The system consists of three main blocks: **Behavior Generation Unit, Decision Unit** and **Performance Evaluation Unit**.

General operating principles of the system can be summarized as follows:

When the system starts operating, it already has some ability to perform the required task in the form of knowledge encapsulated in the fuzzy if-then rules contained within the Behavior Generation Unit. This unit is a slightly modified fuzzy controller. The rules represent the "coarse" knowledge necessary for performing the requested task. The advantage of having a fuzzy rule-base within the structure of the system is two-fold: first, the training time to reach a useful state is avoided; second, the structured representation of the system state makes analysis and understanding of the dynamics easier.

*Ahmet Şekercioğlu is a senior research engineer in the Laboratory for Concurrent Computing Systems, Swinburne University of Technology and his e-mail address is "yas@swin.oz.au".

†Professor Gregory K. Egan is the Director of the Laboratory for Concurrent Computing Systems, Swinburne University of Technology and his e-mail address is "gke@swin.oz.au".

The system processes the sensory information and generates a response. At the same time, sensory input pattern is fed to the Decision Unit and a recalled pattern of "relative rule weightings" is obtained. These "remembered" patterns are used to alter the relative importance of the rules in the Behavior Generation Unit.

The Performance Evaluation Unit measures the performance of the action and generates a scalar value related to the system's performance change. This value is used to alter the response of the system to move from a lower performing state to a higher performing state. If the performance change is higher than a certain threshold, the response is not altered. On the other hand, if the value is low, the response is altered. To enhance the system's response, new sensory input - relative rule strength associations are produced to replace the lower performing associations stored in the Decision Unit.

Similar sensory input patterns are associated with the most successful response pattern, and, long term changes in the environment dynamically modify the associations.

Performance measuring criteria within the Performance Evaluation Unit changes according to the goal and physical structure of the system.

In this study we assume that the definition of the variable's classes or fuzzy sets are reasonable and choose to tune the relative rule strength weightings. It is we believe more likely that errors in rule formulation will occur when the expert's knowledge is being captured. Others [NHW92] have assumed that the rules are correct and tuned the fuzzy set definitions.

Following sections describe the operation and details of the functional units of the experimental system. Then results of the application to the truck back-up problem is presented.

2 Operation of the Heuristic Learning Neural-Fuzzy System

Main functional blocks of the system and their relationship is shown in figure 1.

The system receives information from the outside world via the input lines

$$x_1, x_2, \ldots, x_m.$$

These lines represent the stimuli from the environment. The response of the system is represented by the output lines

$$y_1, y_2, \ldots, y_k.$$

Behavior of the system as a response to a particular pattern of input stimuli is encapsulated in the Behavior Generation Unit. In other words, when the system comes to life, it already "knows" how to perform a certain task under certain circumstances.

When input patterns begin appearing on the input lines $x_1, x_2, \ldots, x_m$ the system starts generating responses to achieve its task. At the same time, the input pattern is fed to the Decision Unit. This pattern causes a recall of an associated pattern of "fuzzy rule base modifiers". These modifiers are responsible for altering the actual response of the system. After the first sequence of responses, the Performance Evaluation Unit commences measurement of the system performance. If the measured performance is below a threshold, the associations stored in the Decision Unit are modified. Since only the associations generating the "fuzzy rule base modifiers" related to the inferior response of the system are modified only, repeated trials of same task lead to better generated performance. Therefore, the Decision Unit makes the associations and "remembers" the successful generalized response patterns against the

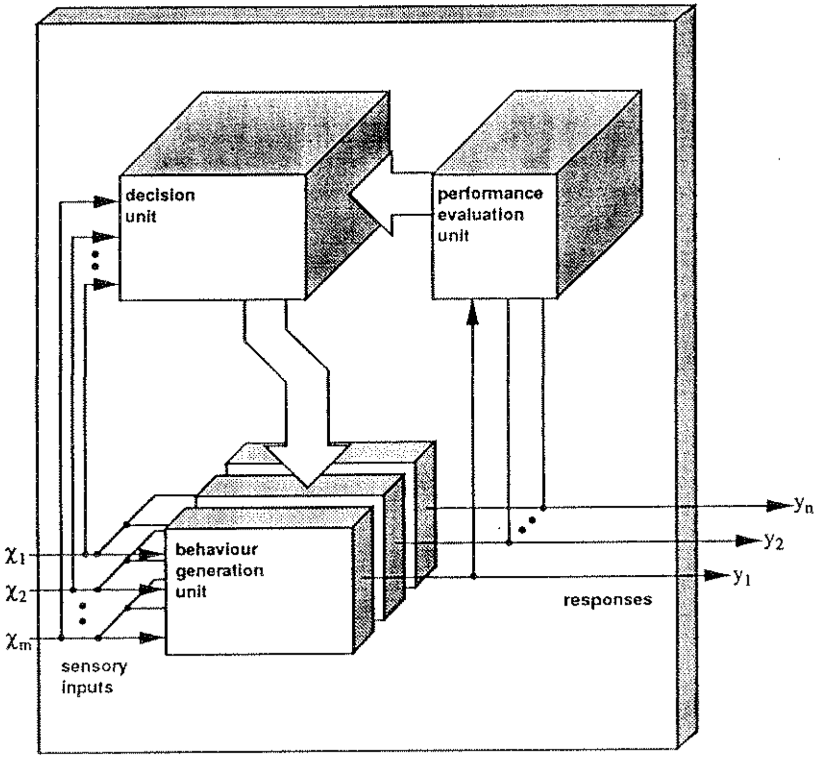

Figure 1: Block diagram of the Heuristic Learning Neural-Fuzzy System.

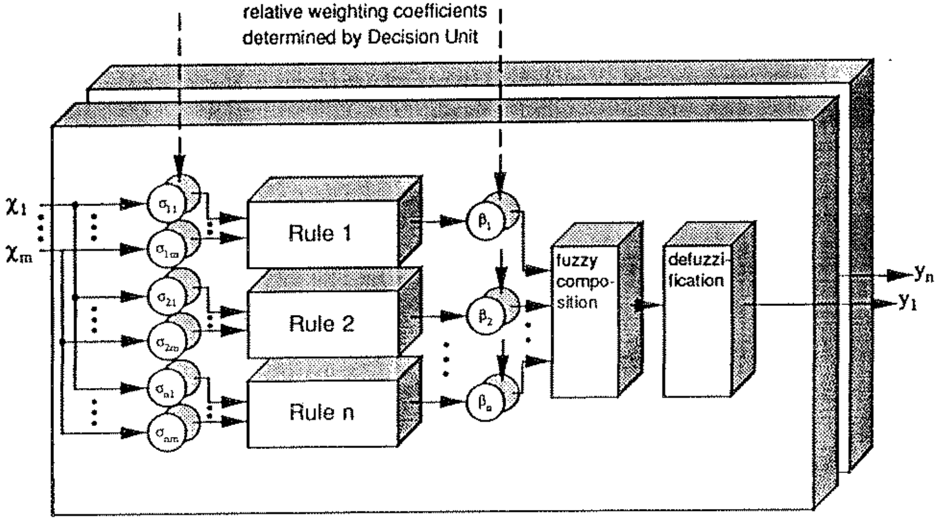

Figure 2: Overall structure of the **Behavior Generation Unit**.

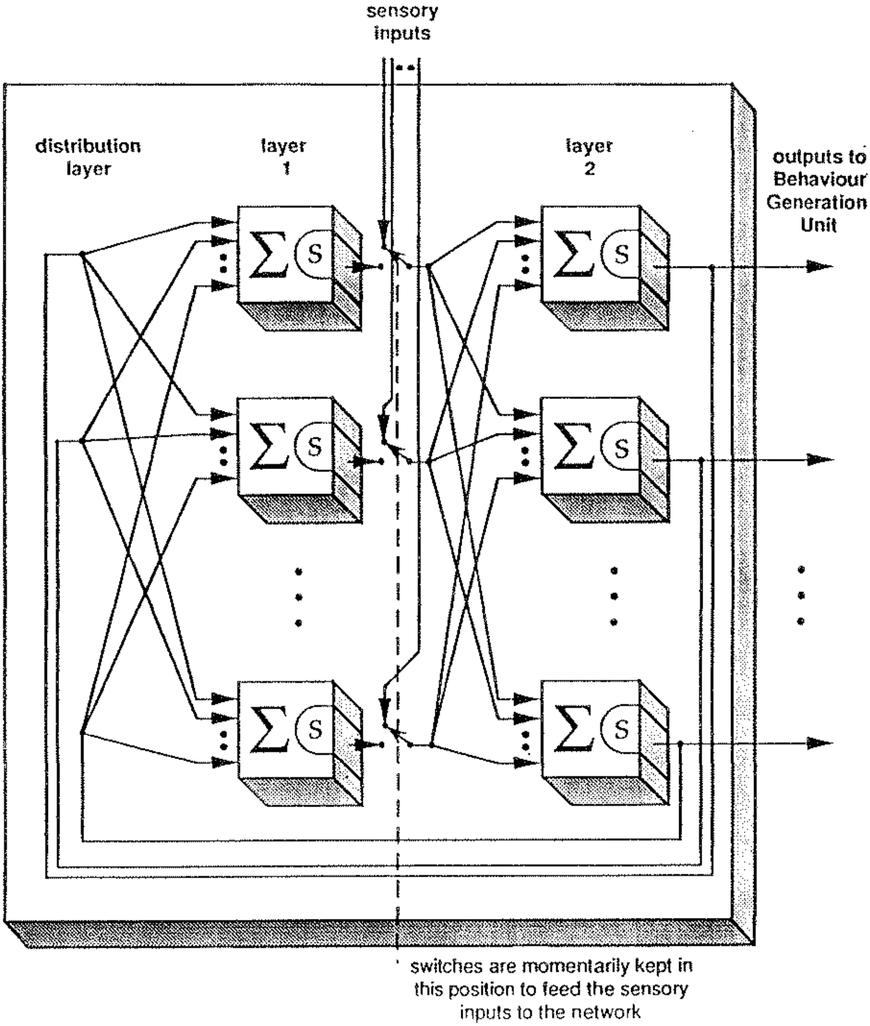

Figure 3: Overall structure of the **Decision Unit**.

various sets of input patterns. After some repeated trials, when the system gets sensory input pattern, it dynamically associates this input pattern with the "fuzzy rule base modifiers" pattern leading to best performance based on its past experience.

It can be expected that, due to the long term changes in many environments, some input pattern combinations will appear less frequently and, will ultimately never reoccur. So, some rules in the Behavior Generation Unit will become less effective.

3 Behavior Generation Unit

Behavior Generation Unit is a fuzzy controller [Zad84] [Zad88] having only slight modifications. Figure 2 depicts the general structure of the Behavior Generation Unit.

Operation of the unit can be summarized as below:
First, linguistic variables, possible linguistic values "term sets" of these linguistic variables represented by fuzzy set membership functions and fuzzy control rules that operate on these rules are determined to model the behavior.

Every linguistic variable has a set of linguistic values and the linguistic values are modelled by the fuzzy sets. At a particular instant, a linguistic variable claims membership in one or more of these linguistic values attached with a "degree of confidence".

Furthermore, the instances of linguistic variables and the decisions arrived at by the individual rules are modified when the system is on-line by multiplying their current values by the coefficients

$$\begin{gathered}\sigma_{11}, \sigma_{12}, \ldots, \sigma_{1m} \\ \sigma_{21}, \sigma_{22}, \ldots, \sigma_{2m} \\ \ldots \\ \ldots \\ \sigma_{n1}, \sigma_{n2}, \ldots, \sigma_{nm}\end{gathered}$$

and

$$\beta_1, \beta_2, \ldots, \beta_n$$

which are determined by the Decision Unit.

The rules are in the form

$$\begin{array}{lll} R^1 & : & \text{if } \sigma_{11}x_1 \text{ is } A_1^1 \text{ and } \sigma_{12}x_2 \text{ is } A_2^1 \text{ and } \ldots \text{ and } \sigma_{1m}x_m \text{ is } A_m^1 \\ & & \text{then response is } C_1 \\ R^2 & : & \text{if } \sigma_{21}x_1 \text{ is } A_1^2 \text{ and } \sigma_{22}x_2 \text{ is } A_2^2 \text{ and } \ldots \text{ and } \sigma_{2m}x_m \text{ is } A_m^2 \\ & & \text{then response is } C_2 \\ \ldots & : & \ldots \\ \ldots & : & \ldots \\ R^n & : & \text{if } \sigma_{n1}x_1 \text{ is } A_1^n \text{ and } \sigma_{n2}x_2 \text{ is } A_2^n \text{ and } \ldots \text{ and } \sigma_{nm}x_m \text{ is } A_m^n \\ & & \text{then response is } C_n \end{array}$$

where

$$\begin{array}{rl} R^i & \text{is the } i^{th} \text{ rule of the rule base,} \\ x_1, x_2, \ldots, x_m & \text{are sensory inputs,} \\ A_j^i & \text{'s are the linguistic values that linguistic variables} \\ & \text{in the rule } R^i \text{ can claim membership and,} \\ C^i & \text{is the response recommended by the rule } R^i. \end{array}$$

For any given sensory input vector $(x_1, x_2, \ldots, x_m)$, the rules individually determine their responses and these responses have confidence values, W^i's attached with them. Each of these confidence values can be calculated as

$$W^i = \mu_{A_1^i}(\sigma_{i1}x_1) \wedge \mu_{A_2^i}(\sigma_{i2}x_2) \wedge \cdots \wedge \mu_{A_m^i}(\sigma_{im}x_m)$$

where $\mu_{A_j^i}(\sigma_{ij}x_j)$ is the grade of membership of a particular value of the linguistic variable x_j scaled by the factor σ_{ij} and fuzzy-and operation denoted by the symbol "$\wedge$" is calculated as the minimum of the grade of memberships

$$W^i = min(\mu_{A_j^i}(\sigma_{ij}x_j)),\ j = 1, 2, \ldots, m.$$

To find the overall response of the system to the sensory inputs $x_1, x_2, \ldots, x_m$, first, in the fuzzy composition stage, the membership function $\mu_{C_i}(y)$ of the the particular linguistic value for the output linguistic variable suggested by each rule is scaled by the "adaptation weight" β_i and confidence level or rule strength, W^i of the rule, such as

$$\tilde{\mu}_{C_i}(y) = W^i\beta_i\mu_{C_i}(y).$$

Second, in the defuzzification stage, the centroid of the areas of the scaled membership functions of the suggested linguistic values is calculated to find the "crisp" value of the response:

$$y_{crisp} = \frac{\sum_i \int \tilde{\mu}_{C_i}(y)ydy}{\sum_i \int \tilde{\mu}_{C_i}(y)dy}.$$

Put simply, every individual rule produces a response having a confidence degree attached to. Furthermore, the coefficients $\beta_1, \beta_2, \ldots, \beta_n$ which are determined by the Decision Unit modify the effect of the individual response to the overall response to reflect the belief of the system at that moment. The resulting combination of the individual responses can be seen as the result of a weighted vote.

4 Decision Unit

The Decision Unit is a "Bidirectional Associative Memory" [Was89] [Kos88]. Bidirectional Associative Memories are two-layer recurrent neural network structures. They are used to encode pattern pairs. They are always stable and provide instant recall of either of the two pattern-pair elements.

To encode an association, say, (A_i, B_i) pair, the correlation matrix

$$C = A_i^T B_i$$

is calculated and added to the weight matrix M.

After encoding the associations, whenever pattern A_i or a sufficiently similar pattern is presented to the Bidirectional Associative Memory, pattern B_i can be recalled.

To erase an association $((A_i, B_i)$,

$$\overline{C} = -A_i^T B_i$$

is calculated and added to the weight matrix M.

The overall structure of the Decision Unit is depicted in figure 3.

5 Performance Evaluation Unit

The functions judging the on-line performance of the system are application specific. In general it is tempting to encapsulate in the performance evaluation function more expertise

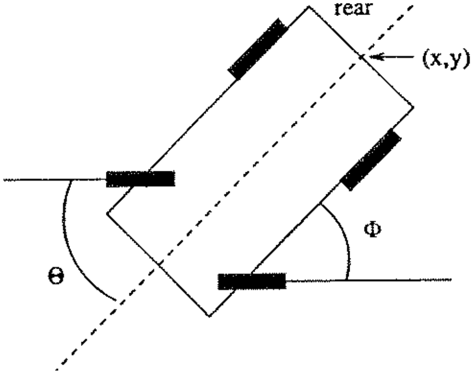

Figure 4: diagram of simulated truck

than the rules themselves. In this study we used a relatively unsophisticated evaluation function to prune a small number of incompetent rules from the rule set, and to tune the remaining rules. The incompetent rules may typify errors by the expert in encoding the rule set or rules which may conflict in some way not necessarily foreseable at the time the rules were encoded.

For our experiment we selected a simple strategy for generating the replacement associations. If the performance is low, we reduce the relative rule weighting of the rule having largest influence on the current decision and increase the relative rule weighting of the rule having smallest (but greater than zero) influence.

6 Application to Truck Back-up Problem

6.1 Problem Description

The truck back-up example is due to [NW89] where an artificial neural network controller is described. The principal reference used by us is due to [KK92] where this example is also used for a fuzzy logic study. Readers are referred to this paper for a more fullsome description of the problem and how the rule set was developed by them. This rule set was used as the starting point for this study.

The problem is to steer a truck from some arbitrary starting position to a loading dock. The input variables for the controller are the angle of the truck to the X-axis (ϕ) and the position of the truck in the X-direction (x). The output variable is the truck steering angle (θ). It is assumed the truck is sufficiently far from the loading dock in the (negative) Y-direction. The loading dock is assumed to be at $(50, 0)$.

The problem and its associated variables is shown in figure 4.

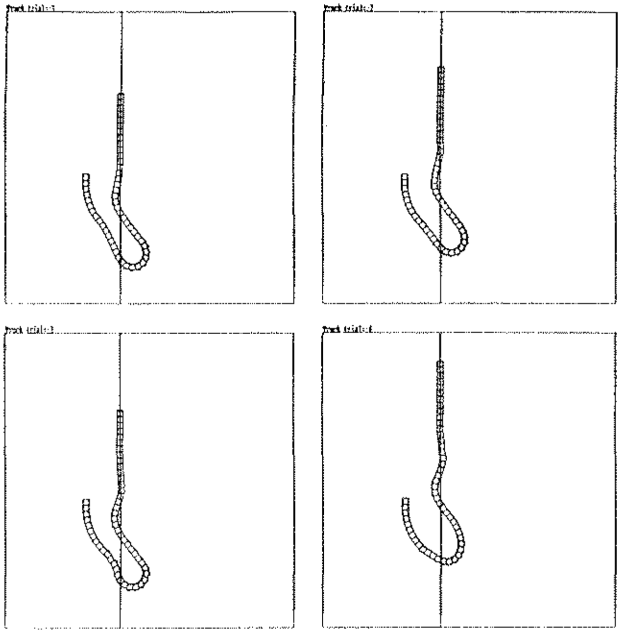

Figure 5: Four consecutive trials of simulation of truck back-up from the initial position $(20.0, -150.0)$ with $\phi = -90.0$ degrees.

6.2 Performance Measure

For the truck back-up example, evaluation of the system performance can be measured by this criteria:

"Ideally the truck should approach the dock at 90 degrees to the x-axis and aligned with the x co-ordinate of the docking point. The performance measure used to evaluate the current rule(s) performance then is as follows:

If the distance from the docking point is increasing and the rate of change of distance is also increasing, or, the angle of approach relative to the ideal approach line is increasing and the rate of change of angle is increasing then the performance is judged to be bad."

6.3 Experiments

We introduced some "bad" rules into the rule set developed by [KK92] to see whether the system will weed out the bad ones after repeated trials. Table 6.3 shows a number of "bad" rules marked in bold face introduced into the rule set and figure 5 shows results of some simulations of backing up the truck from the initial position $(20.0, -150.0)$ with $\phi = -90.0$ degrees.

So far we have obtained limited success. As can be observed from the table 6.3, successive trials reduced the relative rule weightings of the "bad" rules. But, at the same time, relative rule weightings of the other rules also effected.

7 Conclusion and Future Research

So far, our experiments have shown that, the approach we proposed to refine the fuzzy rule bases can be a useful one for complex systems which require large number of fuzzy variables and rules. But, there are still problems waiting to be solved. First problem is, the usefulness of our approach depends heavily on expressing the "performance evaluation function". Finding a performance evaluation function for some complex problems can be difficult and limits the usefulness of our approach. Second limitation is the memory capacity of the Bidirectional Associative Memories. Because of this capacity problem we could not use the coefficients ($\sigma's$) modifying the relative weights of the input variables. To overcome the capacity problem we are planning to conduct experiments using Bidirectional Associative Memory Systems developed by Simpson [Sim90].

References

[KK92] Seong-Gon Kong and Bart Kosko. Adaptive fuzzy systems for backing up a truck and trailer. *IEEE Transactions on Neural Networks*, 3(2):211–223, March 1992.

[Kos88] Bart Kosko. Bidirectional associative memories. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):49–60, January/February 1988.

[NHW92] Hiroyoshi Nomura, Isao Hayashi, and Noboru Wakami. A learning method of fuzzy inference rules by descent method. In *Proceedings of the 1992 IEEE International Conference on Fuzzy Systems*, pages 203–209, San Diego, California, March 1992. IEEE.

[NW89] D. Nguyen and B. Widrow. The truck backer-uppper: An example of self learning in neural networks. In *Proceedings of International Joint Conference on Neural Networks (IJCNN-89)*, volume 2, pages 357–363, June 1989.

[Sim90] Patrick K. Simpson. Associative memory systems. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN-90)*. IEEE, 1990.

[Was89] Philip D. Wasserman. *Neural Computing Theory and Practice*. Van Nostrand Reinhold, 1989.

[Zad84] Lotfi A. Zadeh. Making computers think like people. *IEEE Spectrum*, pages 26–32, August 1984.

[Zad88] Lotfi A. Zadeh. Fuzzy logic. *IEEE Computer*, (4):83–93, Apr 1988.

Rules	trial 1	trial 2	trial 3	trial 4
if (x is le) and (ϕ is rb) then $\theta :=$ ps	1.5	1.6	1.7	1.8
if (x is le) and (ϕ is ru) then $\theta :=$ ns	0.7	0.3	0.3	0.3
if (x is le) and (ϕ is rv) then $\theta :=$ nm	0.5	0.1	0.1	0.1
if (x is le) and (ϕ is ve) then $\theta :=$ nm	0.8	0.5	0.4	0.3
if (x is le) and (ϕ is ve) then $\theta :=$ ps	0.8	0.4	0.4	0.4
if (x is le) and (ϕ is lv) then $\theta :=$ nb	0.6	0.2	0.2	0.2
if (x is le) and (ϕ is lv) then $\theta :=$ ps	0.6	0.2	0.2	0.2
if (x is le) and (ϕ is lu) then $\theta :=$ nb	0.8	0.4	0.4	0.4
if (x is le) and (ϕ is lb) then $\theta :=$ nb	0.9	0.5	0.5	0.5
if (x is lc) and (ϕ is rb) then $\theta :=$ pm	0.6	0.1	0.1	0.1
if (x is lc) and (ϕ is rb) then $\theta :=$ nb	0.5	0.1	0.1	0.1
if (x is lc) and (ϕ is ru) then $\theta :=$ ps	0.8	0.4	0.4	0.4
if (x is lc) and (ϕ is rv) then $\theta :=$ ns	1.0	0.7	0.7	0.5
if (x is lc) and (ϕ is ve) then $\theta :=$ nm	0.8	0.4	0.4	0.4
if (x is lc) and (ϕ is lv) then $\theta :=$ nm	0.5	0.1	0.1	0.1
if (x is lc) and (ϕ is lu) then $\theta :=$ nb	0.8	0.4	0.4	0.4
if (x is lc) and (ϕ is lb) then $\theta :=$ nb	0.4	0.1	0.1	0.1
if (x is ce) and (ϕ is rb) then $\theta :=$ pm	0.4	0.0	0.0	0.0
if (x is ce) and (ϕ is ru) then $\theta :=$ ze	0.6	0.2	0.2	0.2
if (x is ce) and (ϕ is ru) then $\theta :=$ pm	0.7	0.3	0.3	0.3
if (x is ce) and (ϕ is rv) then $\theta :=$ ps	0.3	0.0	0.0	0.0
if (x is ce) and (ϕ is ve) then $\theta :=$ ze	0.3	0.8	1.0	1.4
if (x is ce) and (ϕ is lv) then $\theta :=$ ns	0.8	0.6	0.5	0.4
if (x is ce) and (ϕ is lu) then $\theta :=$ nm	0.8	0.4	0.4	0.4
if (x is ce) and (ϕ is lb) then $\theta :=$ nm	0.7	0.3	0.2	0.2
if (x is rc) and (ϕ is rb) then $\theta :=$ pb	0.4	0.3	0.3	0.3
if (x is rc) and (ϕ is ru) then $\theta :=$ pb	0.4	0.0	0.0	0.0
if (x is rc) and (ϕ is rv) then $\theta :=$ pm	0.8	0.5	0.5	0.3
if (x is rc) and (ϕ is ve) then $\theta :=$ pm	0.9	0.5	0.5	0.3
if (x is rc) and (ϕ is lv) then $\theta :=$ ps	0.8	0.6	0.6	0.4
if (x is rc) and (ϕ is lu) then $\theta :=$ ns	0.8	0.6	0.6	0.4
if (x is rc) and (ϕ is lb) then $\theta :=$ nm	0.6	0.2	0.2	0.2
if (x is ri) and (ϕ is rb) then $\theta :=$ pb	0.6	0.2	0.2	0.2
if (x is ri) and (ϕ is ru) then $\theta :=$ pb	0.8	0.4	0.4	0.4
if (x is ri) and (ϕ is rv) then $\theta :=$ pb	0.9	0.6	0.6	0.4
if (x is ri) and (ϕ is ve) then $\theta :=$ pm	0.4	0.0	0.0	0.0
if (x is ri) and (ϕ is lv) then $\theta :=$ pm	0.5	0.3	0.3	0.3
if (x is ri) and (ϕ is lu) then $\theta :=$ ps	0.6	0.2	0.2	0.2
if (x is ri) and (ϕ is lb) then $\theta :=$ ns	0.4	0.0	0.0	0.0

Table 1: Full rule set with four introduced "bad" rules marked in bold face and values of the relative rule weightings at the end of each of the four sample trials