Techniques for Extracting Rules from Artificial Neural Networks

Rudy Setiono
School of Computing
National University of Singapore
Lower Kent Ridge Road, Singapore 119260, Singapore
{rudys}@comp.nus.edu.sg

1 Motivations:

The Wisconsin Breast Cancer Diagnosis dataset:

- 9 measurements taken from fine needle aspirates of human breast tissues:

  $\mathcal{A}_1$ clump thickness

  $\mathcal{A}_2$ uniformity of cell size

  $\mathcal{A}_3$ uniformity of cell shape

  $\mathcal{A}_4$ marginal adhesion

  $\mathcal{A}_5$ single epithelial cell size

  $\mathcal{A}_6$ bare nuclei

  $\mathcal{A}_7$ bland chromatin

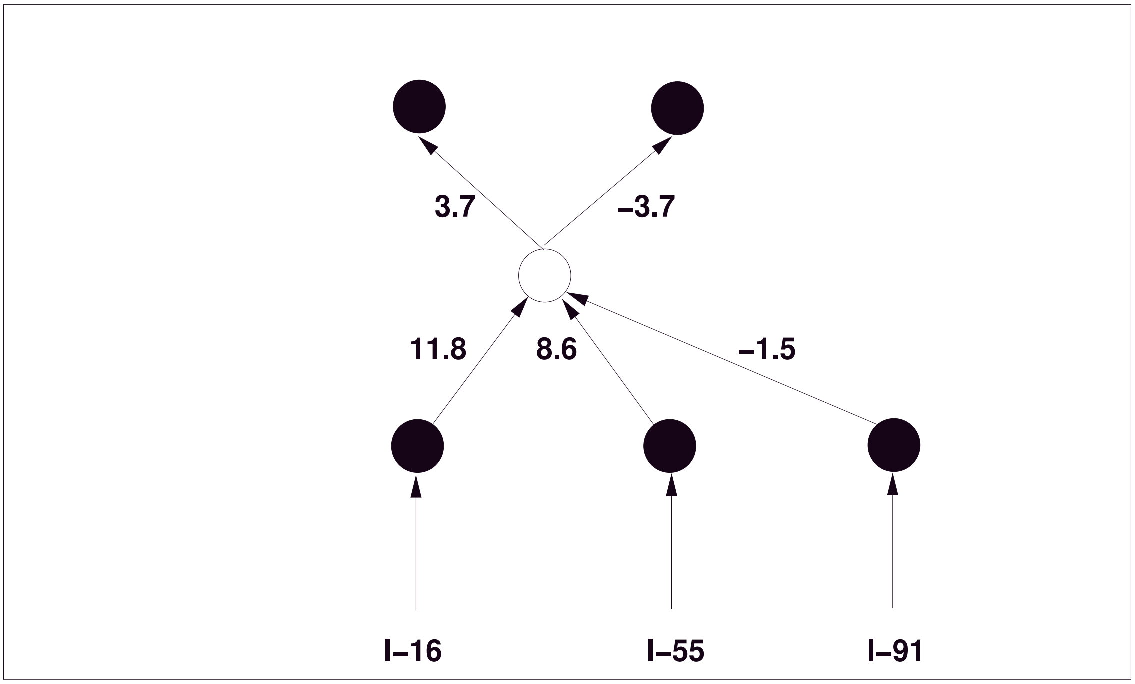
  $\mathcal{A}_8$ normal nucleoli and

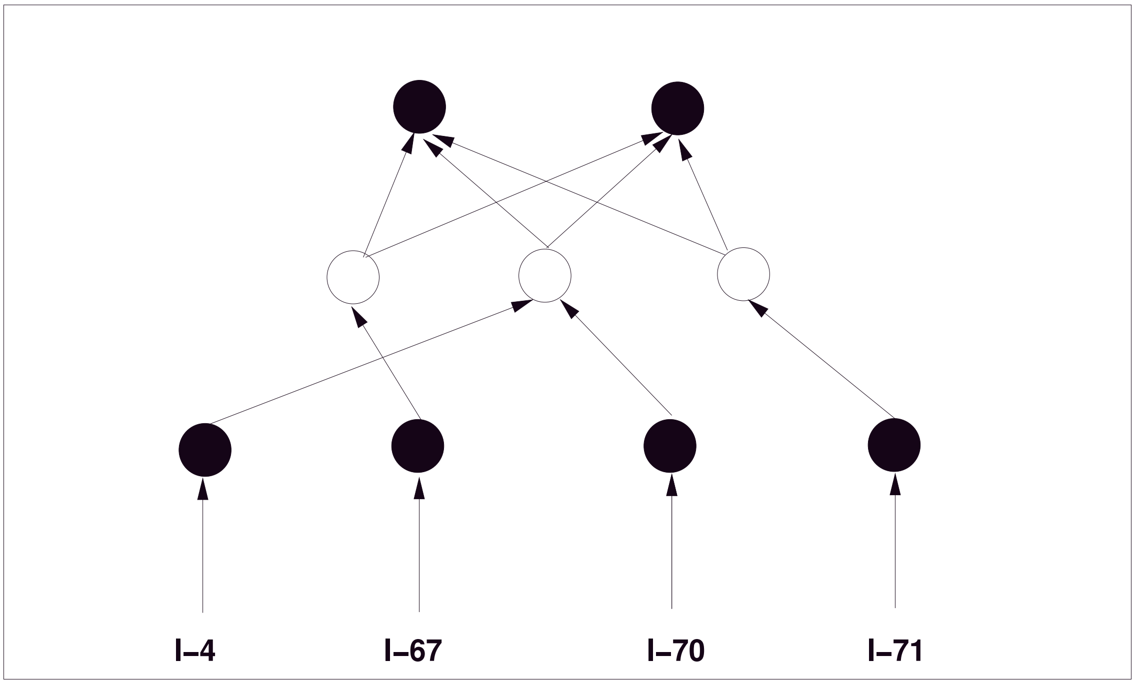
  $\mathcal{A}_9$ mitosis.

- integer-valued ranging from 1 to 10
- 458 are benign samples and 241 are malignant samples
- Average of trained/pruned 100 neural networks (starting with 5 hidden units, 460 connections):

No. connections	10.70	(4.53)	
Accuracy on train data	96.72	(1.11)	%
Accuracy on test data	92.70	(1.57)	%

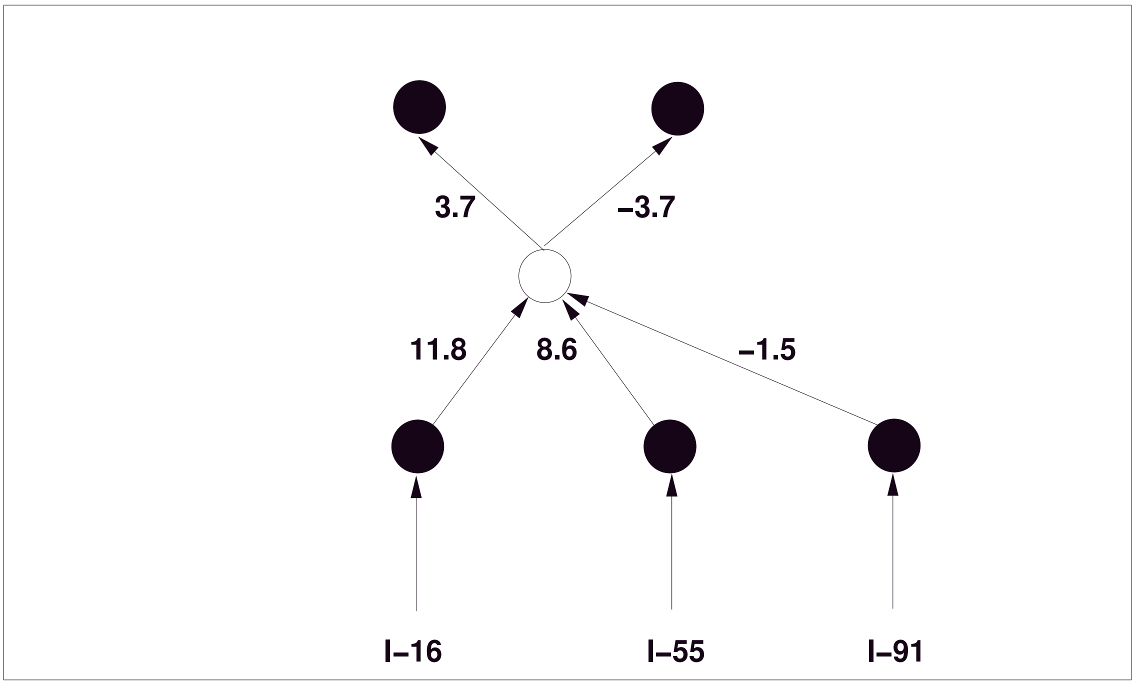
Two pruned networks:





It would be nice to be able to explain the network's classification!

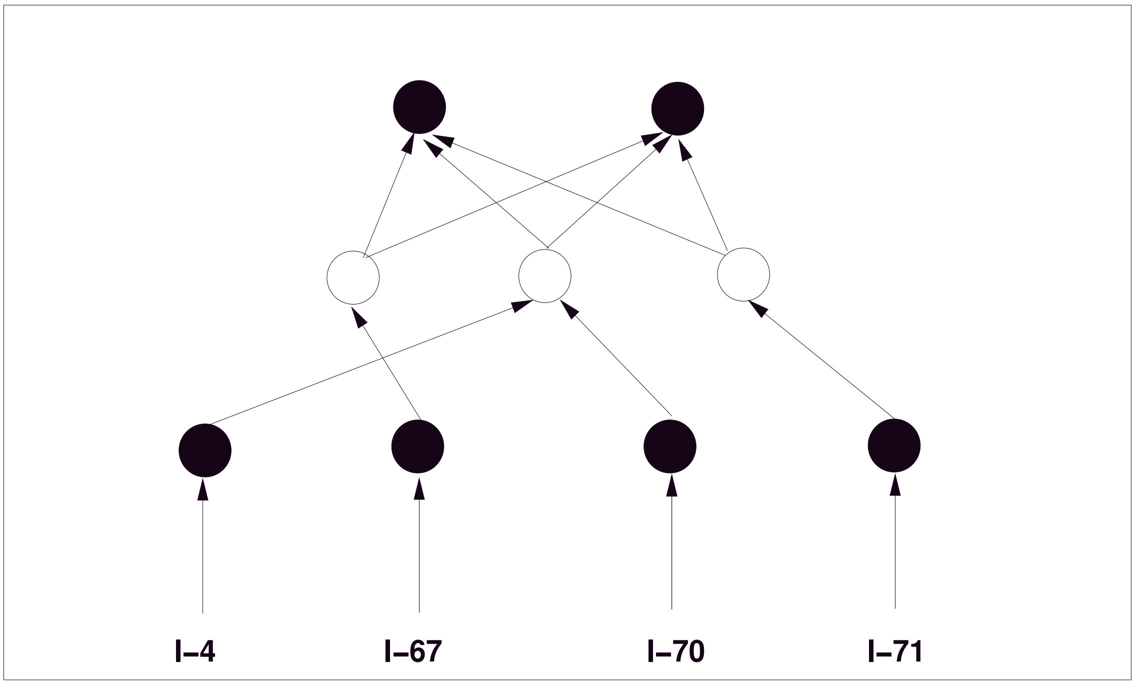
From the first network:



Rule 1:

If $\mathcal{A}_2 \leq 4$ and $\mathcal{A}_6 \leq 5$, then benign.
Else malignant.

From the second network:



Rule 2:

If $\mathcal{A}_1 \leq 6$ and $\mathcal{A}_7 \leq 3$ and $\mathcal{A}_8 \leq 9$, then benign.
Else malignant.

Accuracy of rules = accuracy of corresponding networks.

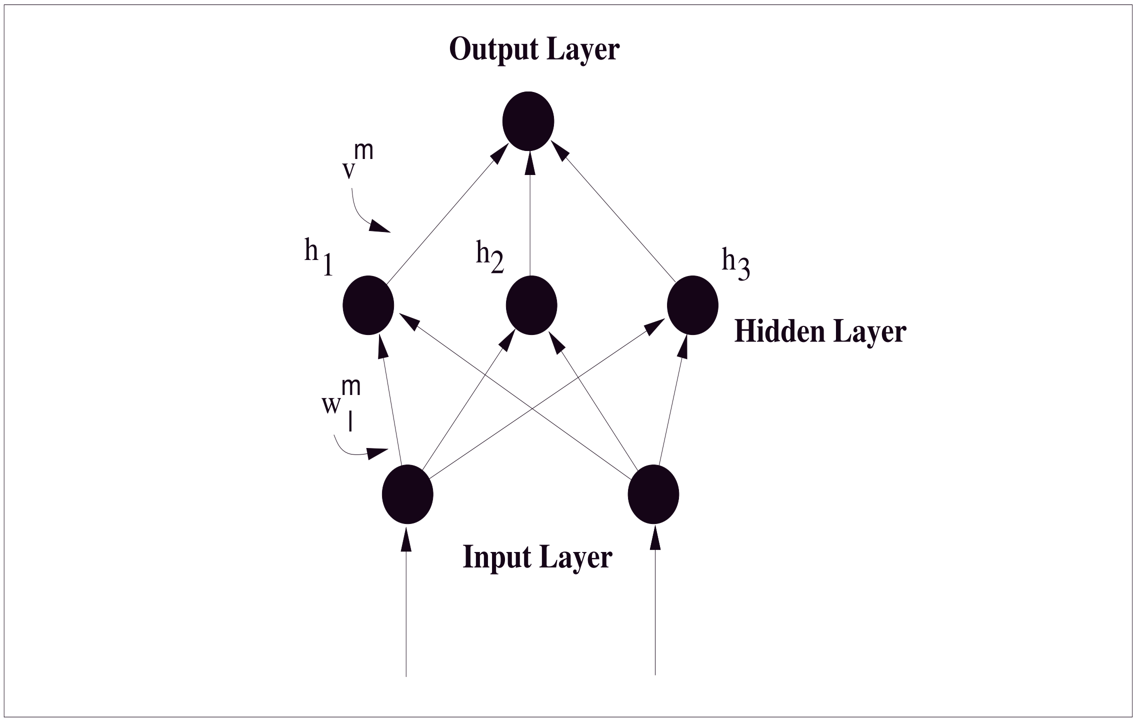
2 Rule extraction from neural networks

- Train and prune a network.
- Cluster hidden unit activation values.
- Generate classification rules in terms of clustered activation values.
- Generate rules to explain the clustered activation values in terms of the input data attributes.
- Merge the two sets of rules.

Network training and pruning.

- Fast training using methods for unconstrained minimization: family of quasi-Newton methods.
- Network error function augmented by a penalty term to facilitate pruning.
- Effective pruning algorithms to remove redundant units and connections.
  Benefits of pruned networks include:
  - less overfitting, higher generalization ability.
  - more concise set of rules.

The neural network:



$$\begin{aligned}
\textbf{min}\ \ \theta(w,v) &= -\sum_{i=1}^{k} t^i \log S^i - (1-t^i)\log(1-S^i) + P(w,v) \\
P(w,v) &= \epsilon_1 \left(\sum_{m=1}^{h} \sum_{\ell=1}^{n} \frac{\beta (w_\ell^m)^2}{1+\beta (w_\ell^m)^2} + \sum_{m=1}^{h} \frac{\beta (v^m)^2}{1+\beta (v^m)^2} \right) + \\
& \quad \epsilon_2 \left(\sum_{m=1}^{h} \sum_{\ell=1}^{n} (w_\ell^m)^2 + \sum_{m=1}^{h} (v^m)^2 \right)
\end{aligned}$$

where

$$\epsilon_1, \epsilon_2, \beta > 0$$

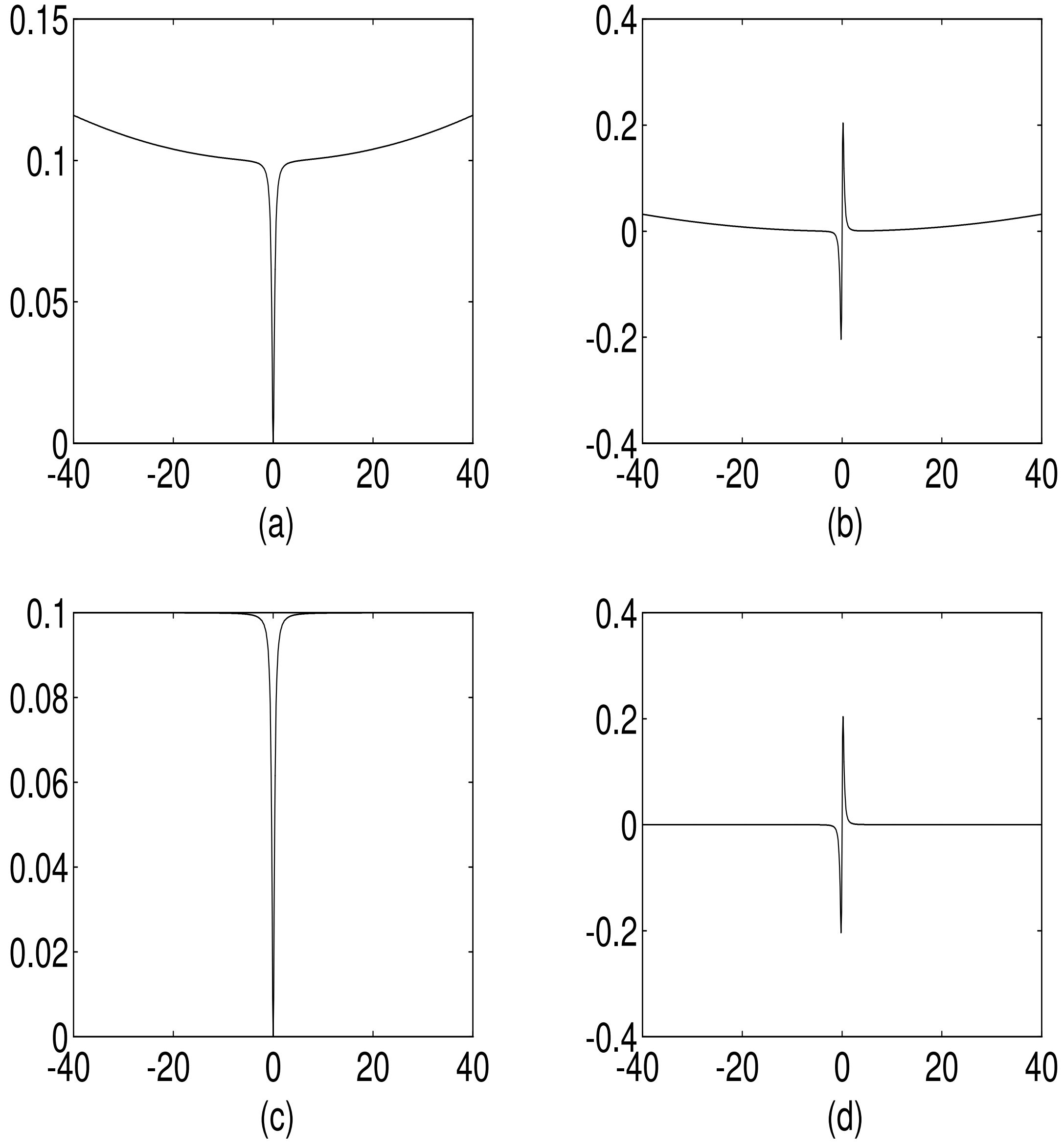
$$S^i = \delta \left(\sum_{j=1}^{h} \delta \left((x^i)^T w^j \right) v^j \right)$$

and

$$t^i = \textbf{ 0 or 1 target value for } x^i$$

Penalty function $f(w) = \epsilon_1 \beta w^2/(1+\beta w^2) + \epsilon_2 w^2$ and its derivative:

- **Figure (a,b).** $\epsilon_1 = 0.1, \epsilon_2 = 10^{-5}$, **and** $\beta = 10$.
- **Figure (c,d).** $\epsilon_1 = 0.1, \epsilon_2 = 10^{-8}$, **and** $\beta = 10$.



Neural network pruning with penalty function (N2P2F) algorithm

1. **Let η_1 and η_2 be positive scalars such that $\eta_1+\eta_2 < 0.5$.**

2. **Pick a fully connected network, and train this network such that error condition is satisfied by all input patterns. Let (w, v) be the weights of this network.**

3. **For each w_ℓ^m, if**
$$|v^m \times w_\ell^m| \leq 4\eta_2, \tag{1}$$
then remove w_ℓ^m from the network

4. **For each v^m, if**
$$|v^m| \leq 4\eta_2, \tag{2}$$
then remove v^m from the network

5. **If no weight satisfies condition (1) or condition (2), then remove w_ℓ^m with the smallest product $|v^m \times w_\ell^m|$.**

6. **Retrain the network. If classification rate of the network falls below an acceptable level, then stop. Otherwise, go to Step 3.**

Clustering of hidden unit activation values.

- **Ch2 algorithm.**
- **Greedy clustering.**
- **C4.5.**

The Chi2 algorithm.

Make use of χ^2 statistic to merge activation values:

$$\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{k} \frac{(H_{ij} - E_{ij})^2}{E_{ij}}$$

where:

k: **the number of classes.**

H_{ij}: **the number of samples in the ith interval, jth class.**

R_i: **the number of samples in the ith interval.**

C_j: **the number of samples in the jth class.**

N: **the total number of samples.**

E_{ij} : **expected frequency of H_{ij}:**

$$E_{ij} = R_i \times C_j / N$$

The algorithm:

1. **Initialize all attributes as "unmarked".**
2. **For each unmarked attribute H$_i$:**
   (a) **Sort the data according to the the activation values of H_i.**
   (b) **Form an initial set of subintervals such that each subinterval contains only one unique value.**
   (c) **For each adjoining pairs of subintervals, compute their χ^2 value.**
   (d) **Find two subintervals with the lowest χ^2 value. If merging the subintervals does not introduce conflicting data, then merge these subintervals, and repeat from Step 2(c). Else if merging the subintervals will introduce conflicting data, mark attribute i.**

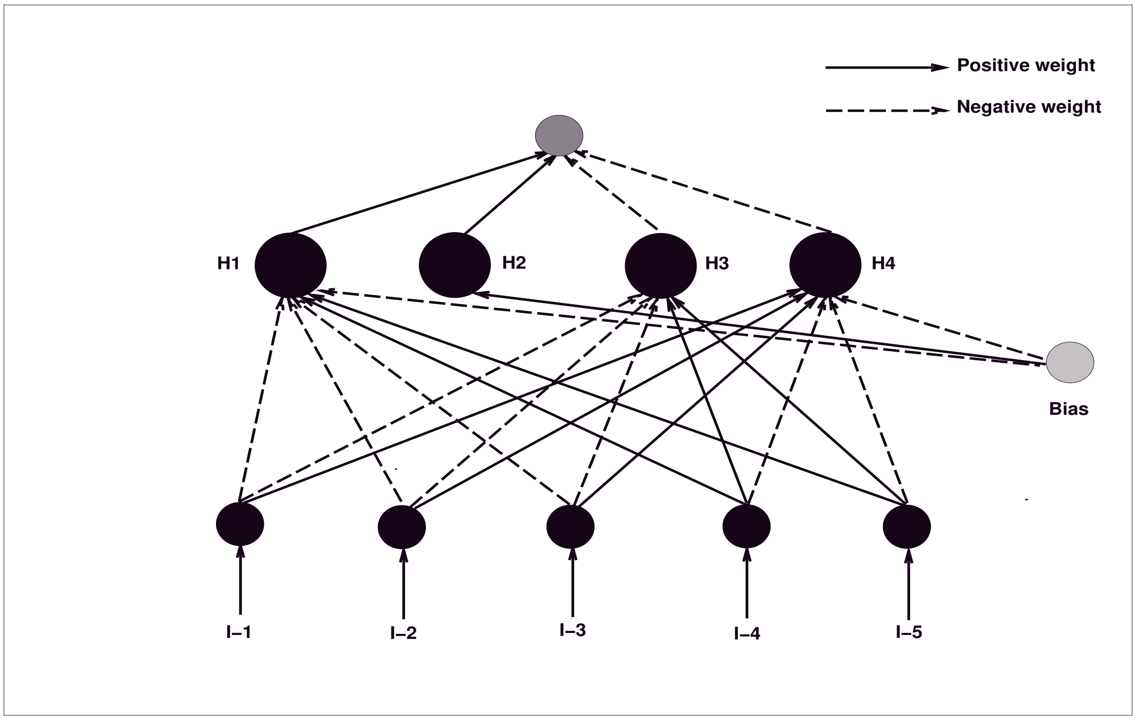
A greedy clustering algorithm (GCA):

1. **Find the smallest positive integer d such that if all the network activation values are rounded to d-decimal-place, the network still retains its accuracy rate.**
2. **Represent each activation value α by the integer closest to $\alpha \times 10^d$. Let $\mathcal{H}_i = < h_{i,1}, h_{i,2}, \ldots, h_{i,k} >$ be the k-dimensional vector of these representations at hidden unit i for patterns $x_1, x_2, \ldots, x_k$ and let $\mathcal{H} = [\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_H]$ be the $k \times H$ matrix of the hidden representations of all patterns at all H hidden units.**
3. **Let P be a permutation of the set $\{1, 2, \ldots, H\}$ and set $m = 1$.**
4. **Set $i = P(m)$.**
5. **Sort the values of the i-th column ($\mathcal{H}_i$) of matrix $\mathcal{H}$ in increasing order.**
6. **Find a pair of distinct adjacent values $h_{i,j}$ and $h_{i,j+1}$ in $\mathcal{H}_i$ such that if $h_{i,j+1}$ is replaced by $h_{i,j}$, no conflicting data will be generated**
7. **If such a pair of values exists, replace all occurrences of $h_{i,j+1}$ in $\mathcal{H}_i$ by $h_{i,j}$ and repeat Step 6. Otherwise, set $m = m + 1$. If $m \leq H$, go to Step 4, else Stop.**

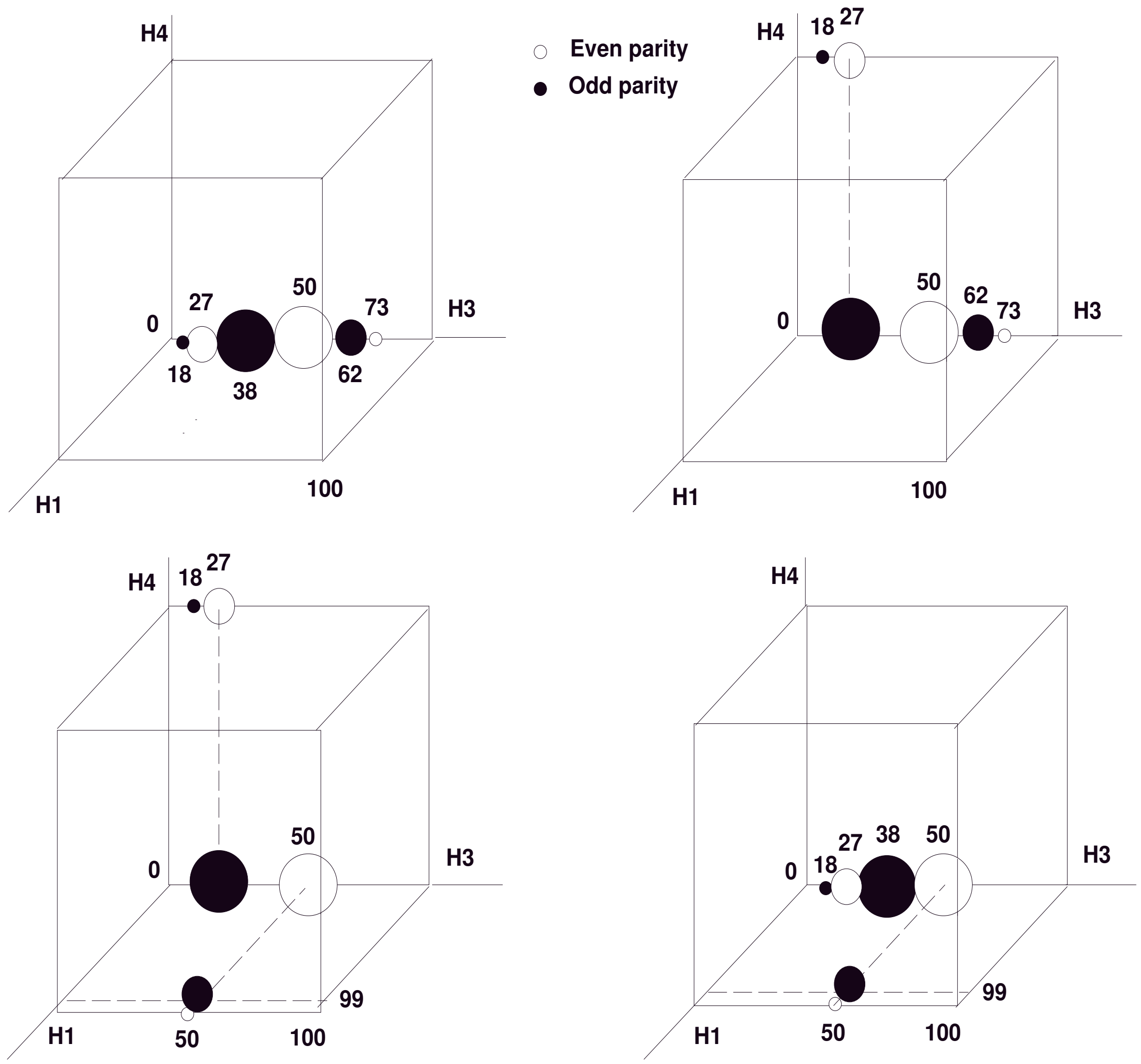
The 5-bit parity problem.

- Input: binary vectors of length 5.
- Output: 1 if the number of one's in the input vector is odd, zero otherwise.

One pruned network:



Different orderings produce different clusters:

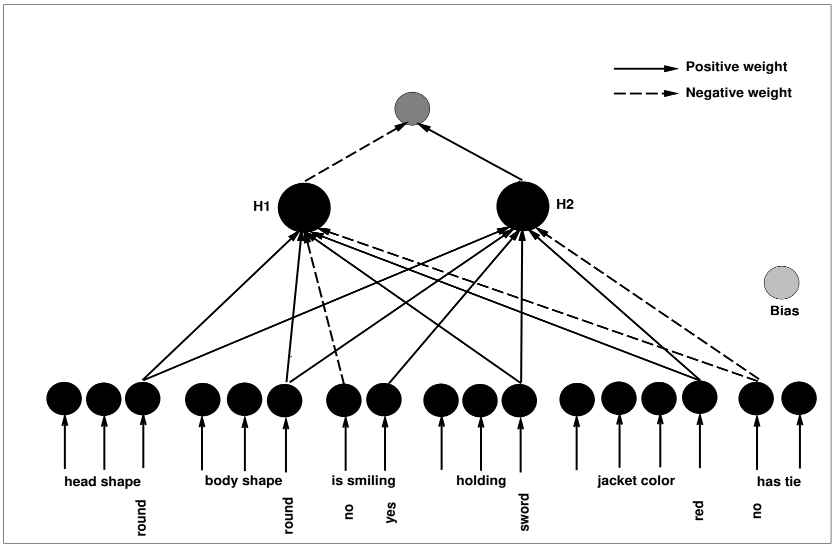


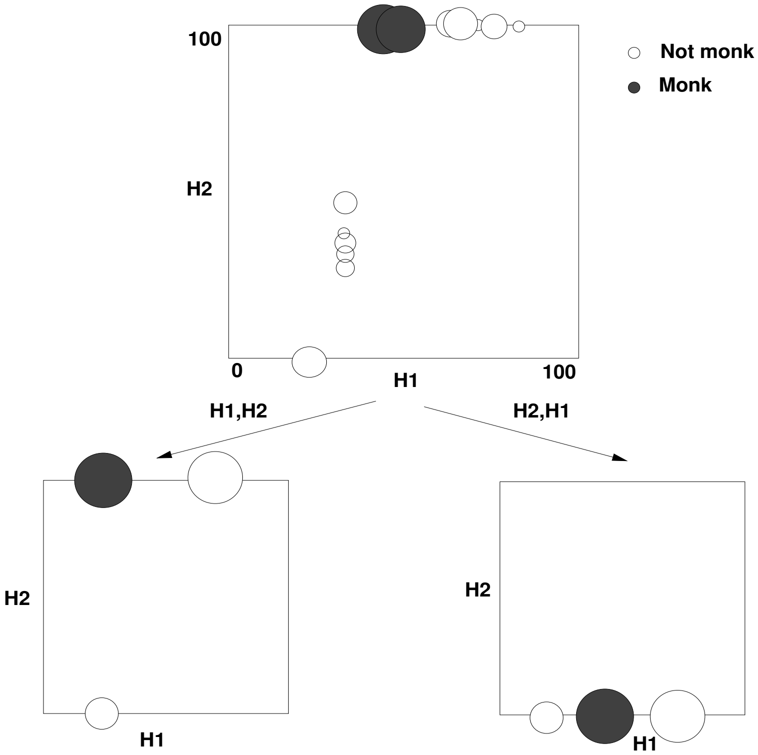
- Large circle = 10 patterns
- Medium circle = 5 patterns
- Small circle = 1 pattern.

The monks problems.
Input attributes:

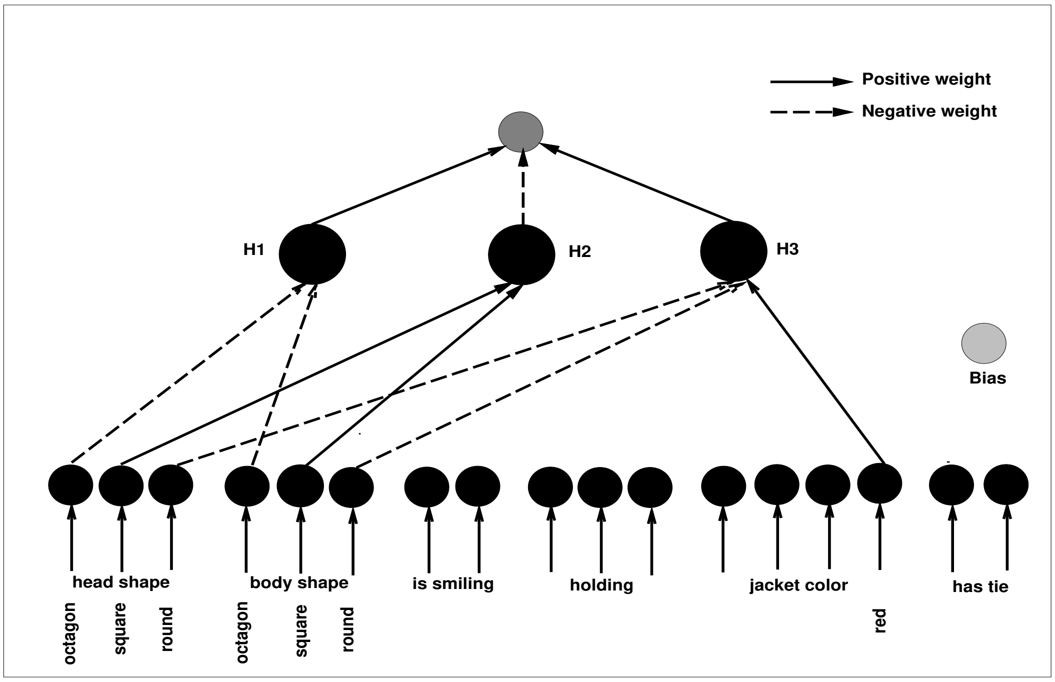
- $\mathcal{A}_1$: **head_shape** $\in$ **round, square, octagon**
- $\mathcal{A}_2$: **body_shape** $\in$ **round, square, octagon**
- $\mathcal{A}_3$: **is_smiling** $\in$ **yes, no**
- $\mathcal{A}_4$: **holding** $\in$ **sword, balloon, flag**
- $\mathcal{A}_5$: **jacket_color** $\in$ **red, yellow, green, blue**
- $\mathcal{A}_6$: **has_tie** $\in$ **yes, no**

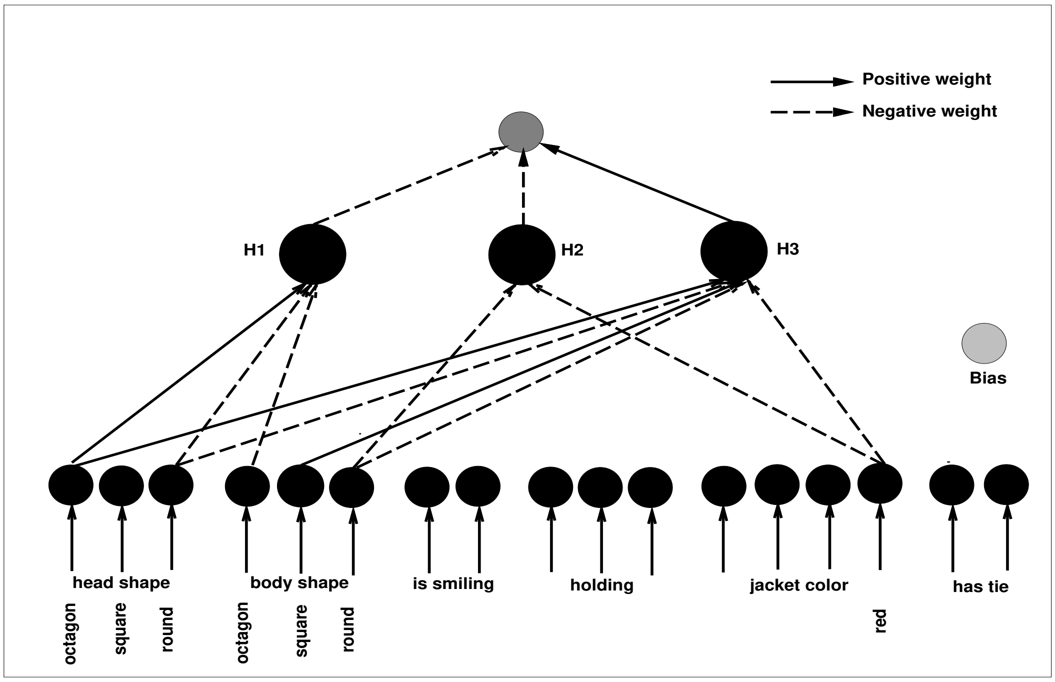
Problem Monks 2:
Exactly 2 of the 6 attributes have their first value.





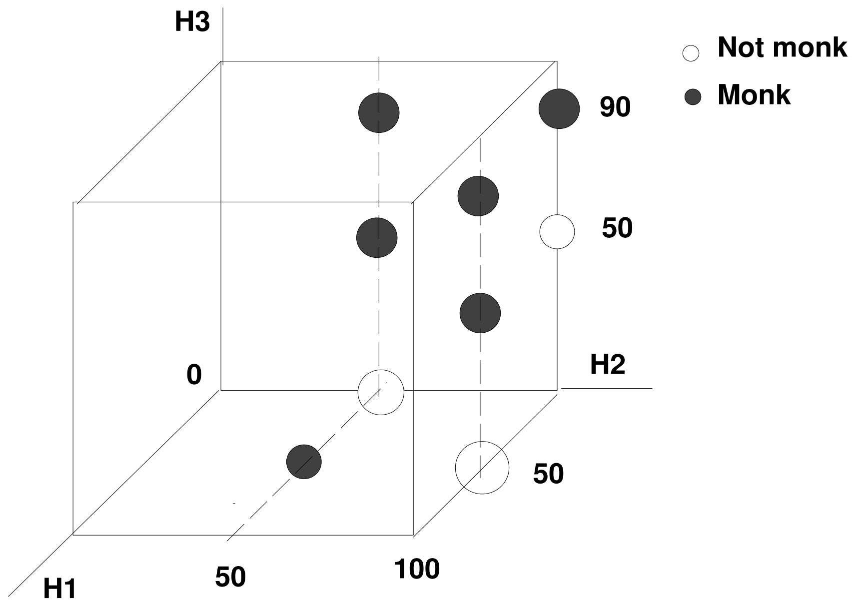
Problem Monks 1:
(head_shape = body_shape) or (jacket_color = red).



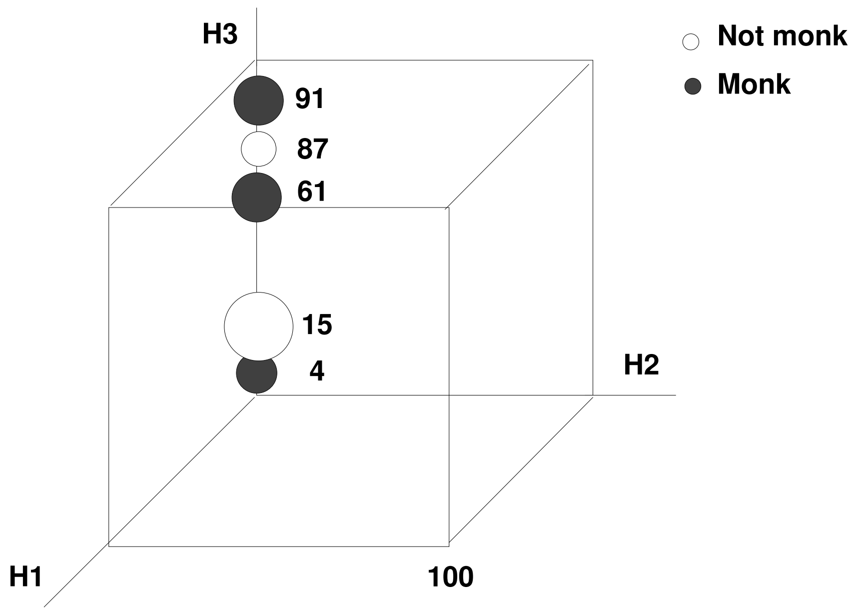


Clustered activation values.

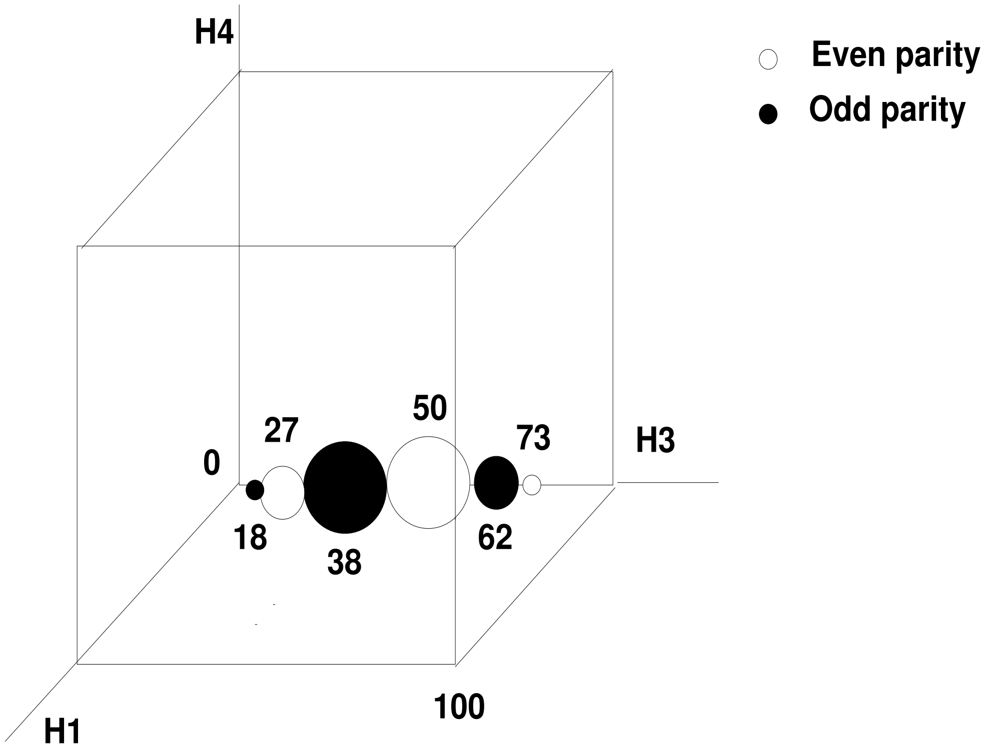
From the first network:



From the second network:



Classification rules in terms of clustered activation values. The 5-bit parity problem:



Rule 1: **If the activation value is in the interval** $[0.18, 0.265)$, **then odd parity.**

Rule 2: **If the activation value is in the interval** $[0.265, 0.375)$, **then even parity.**

Rule 3: **If the activation value is in the interval** $[0.375, 0.495)$, **then odd parity.**

Rule 4: **If the activation value is in the interval** $[0.495, 0.615)$, **then even parity.**

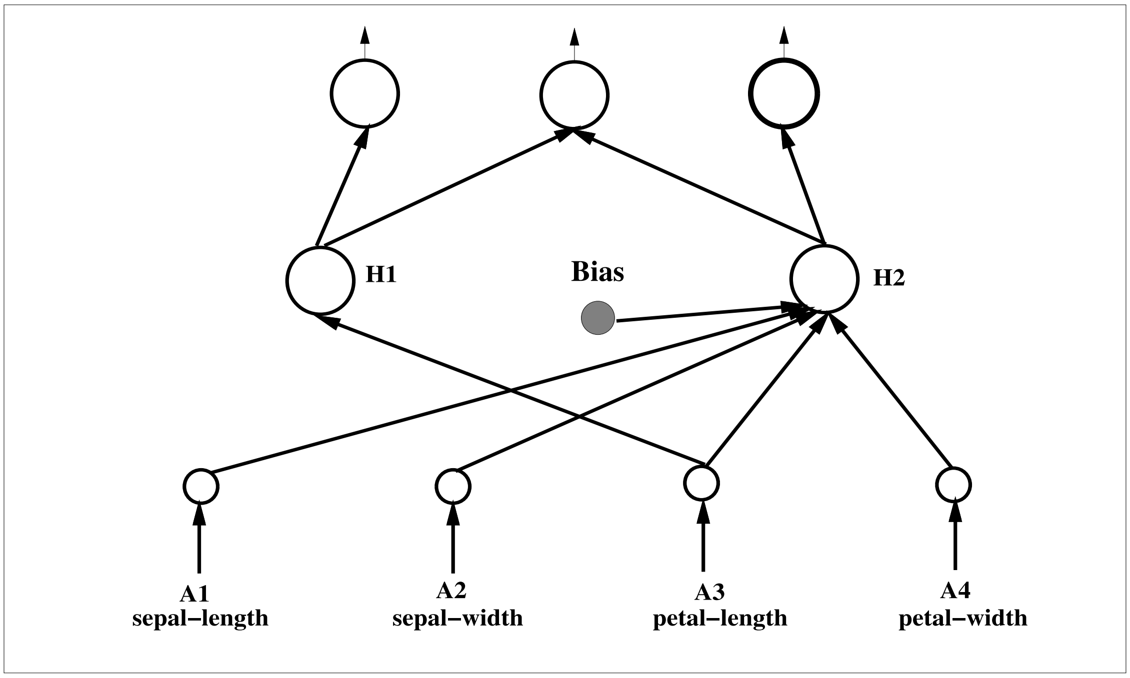
Rule 5: **If the activation value is in the interval** $[0.615, 0.725)$, **then odd parity.**

Rule 6: **If the activation value is in the interval** $[0.725, 1]$, **then even parity.**

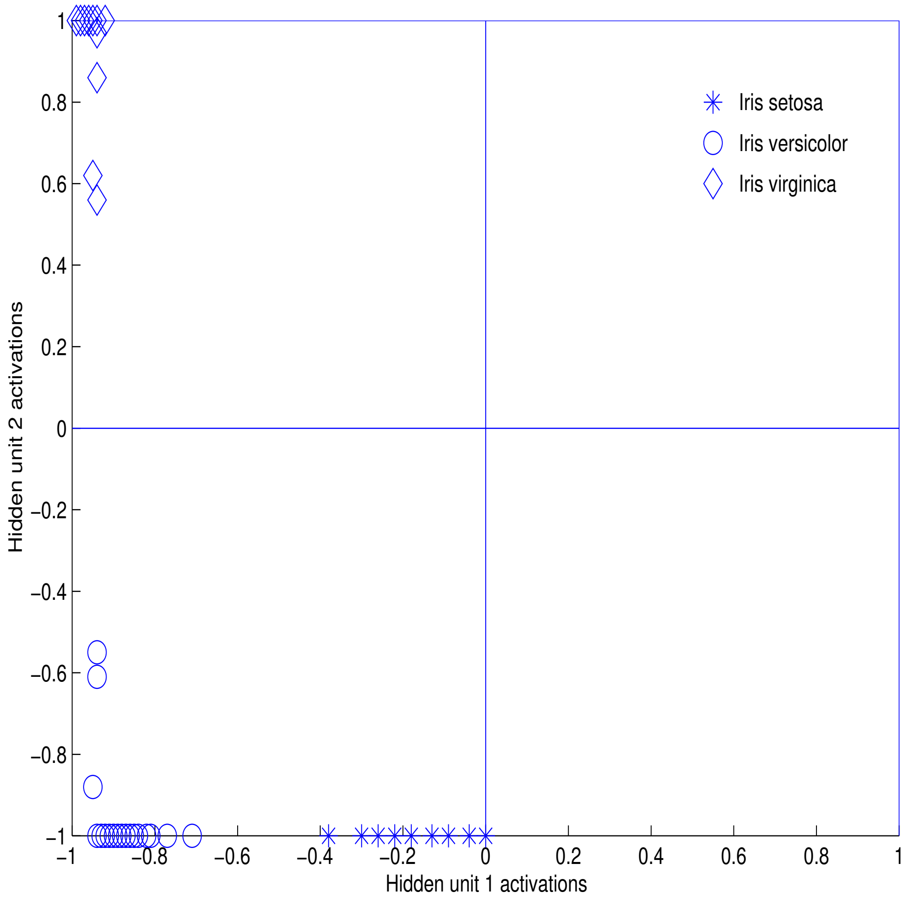
Iris dataset:

- **150 instances**
- **4 continuous attributes:**
  - **sepal_length ($\mathcal{A}_1$)**
  - **sepal_width ($\mathcal{A}_2$)**
  - **petal_length ($\mathcal{A}_3$)**
  - **petal_width ($\mathcal{A}_4$)**
- **Each pattern belongs to one of the 3 possible classes: setosa, versicolor, or virginica.**

A pruned neural network:



The 2-dimensional hidden activation values:



Classification rules:

- **If** $H1 > -0.71$ **: Iris setosa**
- **else if** $H2 <= -0.55$ **: Iris versicolor**
- **else : Iris virginica**

DNF9b Dataset:

Class 1 $x_1x_2x_3 + \overline{x}_1\overline{x}_2 + x_7x_8x_9 + \overline{x}_7\overline{x}_9$,

Class 2 Otherwise.

C4.5 decision tree built using network hidden activations as input:

```
H3 <= 0 : 1
H3 > 0 :
|   H4 > 0.30 : 1
|   H4 <= 0.30 :
|   |   H2 <= -0.07 : 0
|   |   H2 > -0.07 :
|   |   |   H1 <= -0.61 : 0
|   |   |   H1 > -0.61 : 1
```

An equivalent tree where the left hand side of each condition is a linear inequality involving the original input features.

```
H*3 <= 0 : 1
H*3 > 0 :
|   H*4 > 0.31 : 1
|   H*4 <= 0.31 :
|   |   H*2 <= -0.07 : 0
|   |   H*2 > -0.07 :
|   |   |   H*1 <= -0.70 : 0
|   |   |   H*1 > -0.70 : 1
```

Replacing the conditions by the original inputs.

H*3. The splitting condition is

$$3.70x_7 + 3.70x_9 \le 0$$

Clearly, H*3 ≤ 0 if and only if $x_7 = x_9 = 0$.

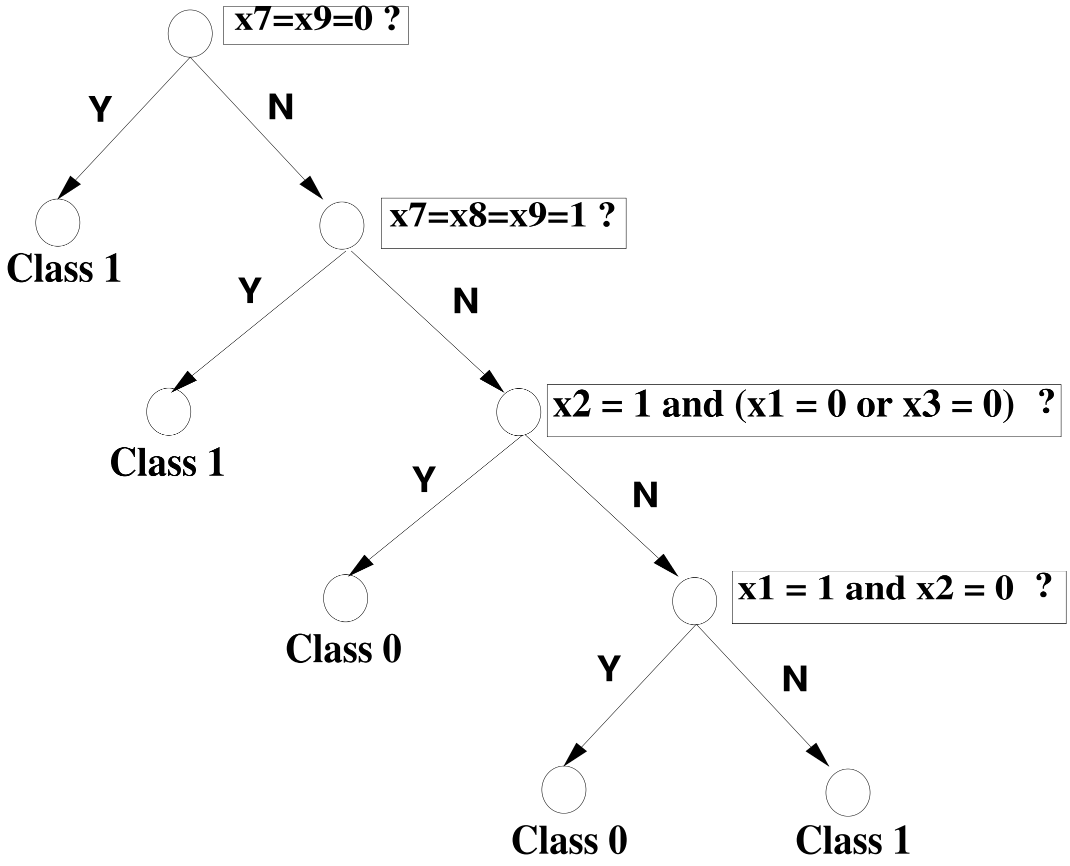
H*4. The splitting condition is

$$0.15x_7 + 0.15x_8 + 0.15x_9 > ge0.31$$

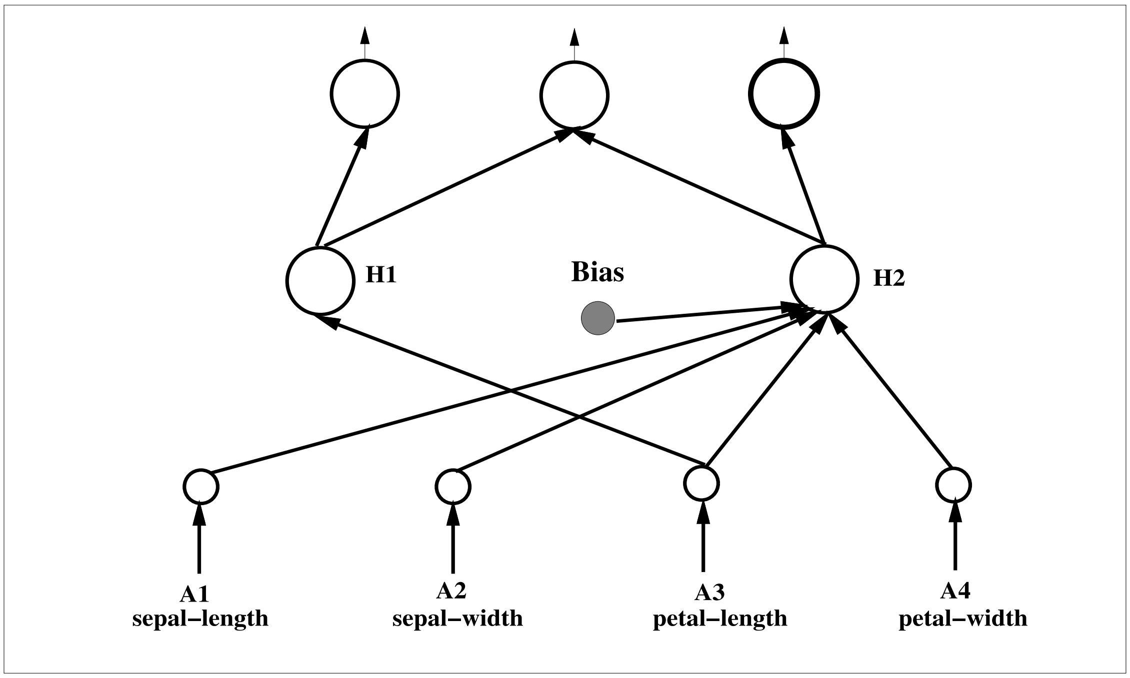
It is true that H*4 > 0.31 if and only if $x_7 = x_8 = x_9 = 1$.

Class 1 $x_1x_2x_3 + \overline{x}_1\overline{x}_2 + x_7x_8x_9 + \overline{x}_7\overline{x}_9$,

Class 2 Otherwise.



Iris dataset:



- **If** $H1 > -0.71$ **: Iris setosa**
- **else if** $H2 <= -0.55$ **: Iris versicolor**
- **else : Iris virginica**

Equivalent set of rules:

- **If** $petal\_length > 2.23$**, then Iris setosa.**
- **Else if** $-\ sepal\_length - 1.57\ sepal\_width + 3.57\ petal\_length + 3.56\ petal\_width > 12.63$**, then Iris versicolor.**
- **Else Iris virginica.**

3 Results of experiments

Experiments with UCI datasets (10-fold cross validation).

Name	#Data	#A	Type
1. Monk1	432	17	binary
2. Monk2	432	17	binary
3. Monk3	432	17	binary
4. TicTacToe	958	27	binary
5. Vote	300	48	binary
6. BreastCancer	699	9	continuous
7. Bupa	345	6	continuous
8. Ionosphere	351	34	continuous
9. Pima-diabetes	768	8	continuous
10. Sonar	208	60	continuous
11. Australian	690	14	mixed
12. HeartDisease	297	13	mixed
13. Housing	506	13	mixed

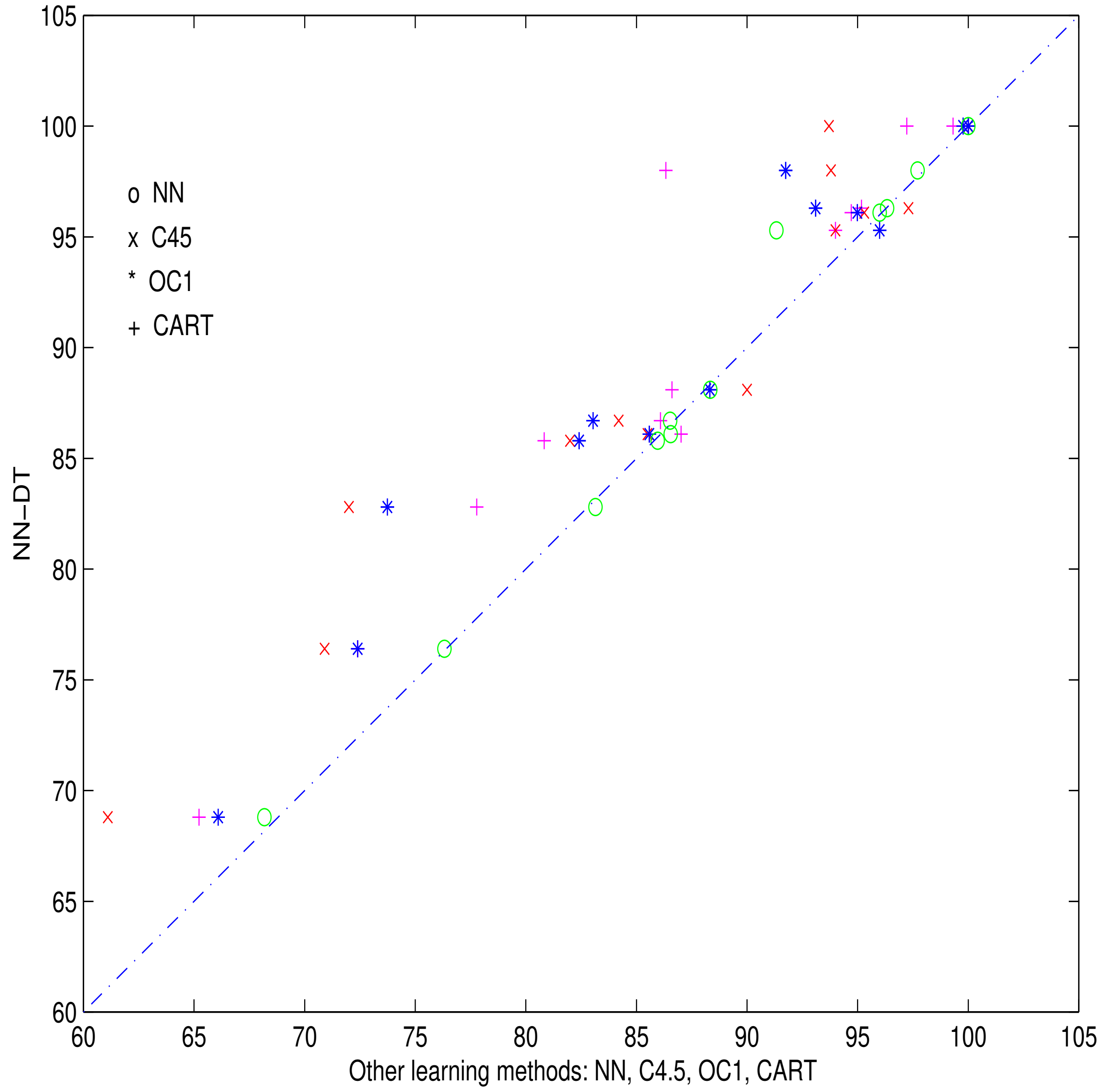
Accuracy:

Name	NN-C4.5	NN	C4.5	OC1	CART
Monk1	100	100	100	100	100
(P-value)		-	-	-	-
Monk2	100	100	93.7	99.77	99.31
(P-value)		-	(**0.0025**)	(0.3434)	(0.3434)
Monk3	100	100	100	100	97.22
(P-value)		-	-	-	(**0.0133**)
TicTacToe	98.0	97.71	93.8	91.75	86.33
(P-value)		(0.1801)	(**0.0001**)	(**0.0034**)	(**0.0001**)
Vote	96.3	96.33	97.3	93.10	95.17
(P-value)		(0.7583)	(0.1944)	(**0.0422**)	(0.4073)
B-Cancer	96.1	96.0	95.3	94.99	94.71
(P-value)		(0.7583)	(0.2423)	(0.4033)	(0.2777)
Bupa	68.8	68.18	61.1	66.09	65.22
(P-value)		(0.5565)	(**0.0475**)	(0.5289)	(0.3443)
Ionosphere	88.1	88.34	90.0	88.32	86.61
(P-value)		(0.5986)	(0.4249)	(0.9376)	(0.6827)
Pima	76.4	76.32	70.9	72.40	72.40
(P-value)		(0.8996)	(**0.0094**)	(**0.0362**)	(**0.0362**)
Sonar	86.1	86.55	85.5	85.58	87.02
(P-value)		(0.3521)	(0.8519)	(0.9524)	(0.8482)
Australian	86.7	85.52	84.2	83.04	86.09
(P-value)		0.3955	(0.2014)	(**0.0922**)	(0.7988)
Heart	82.8	83.15	72.0	73.74	77.78
(P-value)		0.7944	(**0.0046**)	(**0.0068**)	0.1335
Housing	85.8	85.96	82.0	82.41	80.83
(P-value)		0.5912	(**0.0256**)	(0.2625)	(**0.0470**)

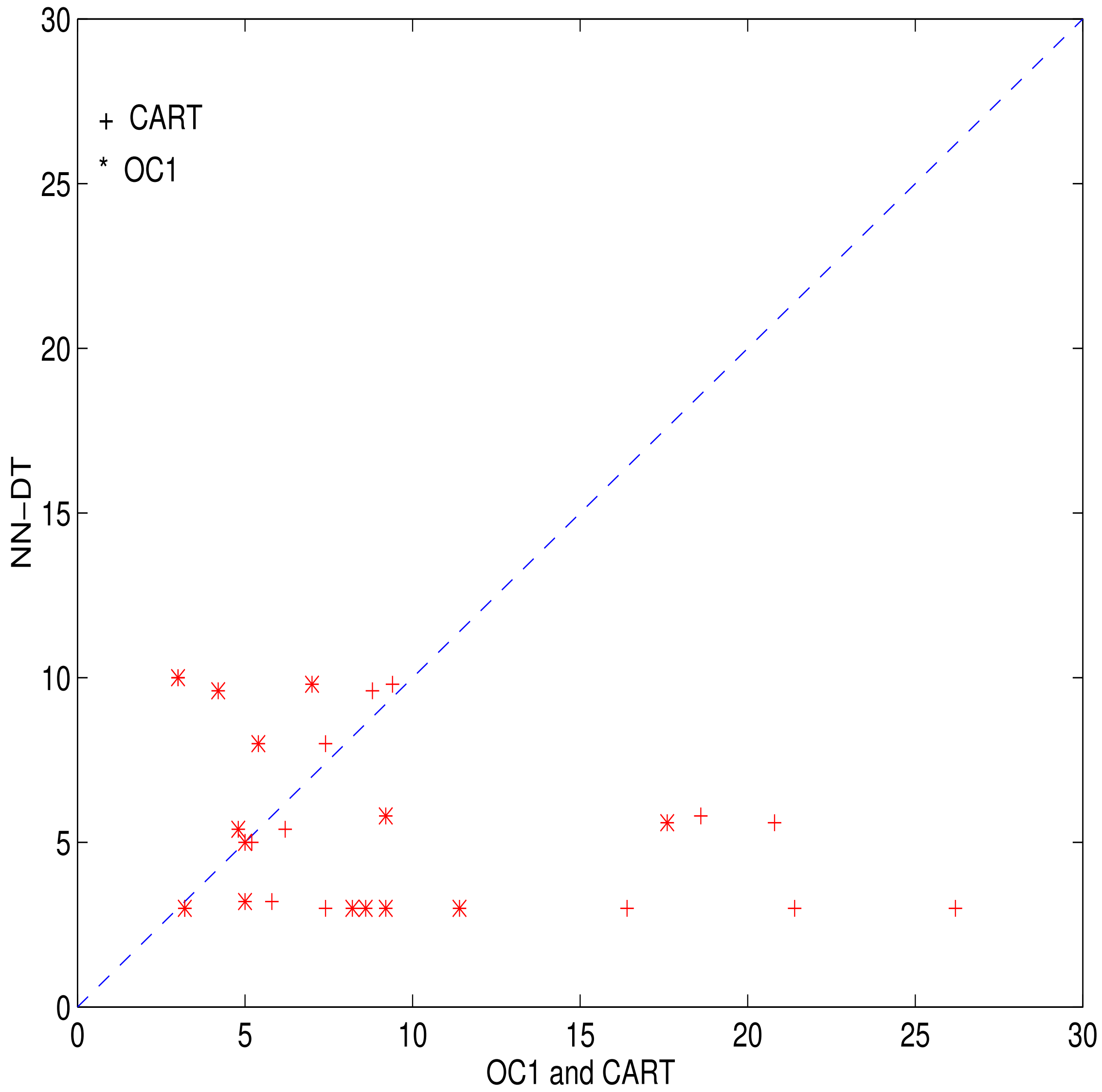
Tree size:

Name	NN-C4.5	C4.5	OC1	CART
Monk1	9.8	14.8	7.0	9.4
(P-value)		(**0.0001**)	(**0.0013**)	(0.5594)
Monk2	8.0	87.6	5.4	7.4
(P-value)		(**0.0001**)	(0.1174)	(0.7418)
Monk3	10.0	15.0	3.0	3.0
(P-value)		(**0.0001**)	(**0.0001**)	(**0.0001**)
TicTacToe	5.6	71.8	17.6	20.8
(P-value)		(**0.0001**)	(**0.0787**)	(**0.0185**)
Vote	3.0	6.6	3.2	3.2
(P-value)		(**0.0001**)	(0.3434)	(0.3434)
B-Cancer	3.2	22.2	5.0	5.8
(P-value)		(**0.0001**)	(0.1105)	(**0.0533**)
Bupa	5.8	79.2	9.2	18.6
(P-value)		(**0.0001**)	(0.3961)	(0.1495)
Ionosphere	3.0	24.8	8.6	21.4
(P-value)		(**0.0001**)	(**0.0552**)	(**0.0005**)
Pima	3.0	122.4	11.4	26.2
(P-value)		(**0.0001**)	(**0.0196**)	(**0.0722**)
Sonar	3.0	31.0	8.2	16.4
(P-value)		(**0.0001**)	(**0.0056**)	(**0.0002**)
Australian	5.4	68.7	4.8	6.2
(P-value)		(**0.0001**)	(0.8226)	(0.7571)
Heart	3.0	51.9	9.2	7.4
(P-value)		(**0.0001**)	(**0.0566**)	(**0.0008**)
Housing	9.6	53.8	4.2	8.8
(P-value)		(**0.0001**)	(0.1510)	(0.8640)

Comparison of accuracy rates of C4.5, OC1, and CART against NN-C4.5's.



Comparison of tree sizes of OC1 and CART against NN-C4.5's.



Artificial datasets with 9 or 12 binary attributes.

Problem	Accuracy (%)			Tree size	
	NN	C4.5	NN-C4.5	C4.5	NN-C4.5
DNF9a	100.0	100.0	100.0	15.0	5.0
DNF9b	100.0	99.8	99.9	37.0	13.4
DNF12a	100.0	100.0	100.0	31.8	9.0
DNF12b	100.0	100.0	99.9	41.8	21.6
CNF9a	100.0	100.0	100.0	15.0	5.0
CNF9b	100.0	100.0	100.0	17.4	9.0
CNF12a	100.0	100.0	100.0	41.0	7.8
CNF12b	99.2	99.2	99.3	107.4	35.8
MAJ9a	100.0	88.5	100.0	7.0	3.0
MAJ9b	100.0	63.1	100.0	44.4	3.0
MAJ12a	100.0	93.4	100.0	29.0	3.0
MAJ12b	100.0	74.2	100.0	223.4	3.0
PAR9a	100.0	100.0	100.0	15.0	7.0
PAR9b	98.6	90.3	98.2	115.4	11.0
PAR12a	100.0	100.0	100.0	31.0	11.0
PAR12b	99.3	92.3	99.5	484.6	24.2
MUX9	100.0	100.0	100.0	42.6	11.8
MUX12	100.0	100.0	99.9	210.8	22.4

4 Conclusion

Neural networks can be an alternative tool to decision tree methods for generating symbolic rules.

1. **Neural networks consider the contribution of the inputs on the classification as a group.**
2. **Neural networks are capable of generating oblique hyperplanes.**

DNF9b Dataset:

Class 1 $x_1x_2x_3 + \overline{x}_1\overline{x}_2 + x_7x_8x_9 + \overline{x}_7\overline{x}_9$,

Class 2 Otherwise.

C4.5 decision tree:

```
A1 = 0:
|   A2 = 0: 1 (128.0)
|   A2 = 1:
|   |   (ST1)
|   |   A7 = 0:
|   |   |   A9 = 0: 1 (32.0)
|   |   |   A9 = 1: 0 (32.0)
|   |   A7 = 1:
|   |   |   A8 = 0: 0 (32.0)
|   |   |   A8 = 1:
|   |   |   |   A9 = 0: 0 (16.0)
|   |   |   |   A9 = 1: 1 (16.0)
A1 = 1:
|   A2 = 0:
|   |   (ST2)
|   |   A7 = 0:
|   |   |   A9 = 0: 1 (32.0)
|   |   |   A9 = 1: 0 (32.0)
|   |   A7 = 1:
|   |   |   A8 = 0: 0 (32.0)
|   |   |   A8 = 1:
|   |   |   |   A9 = 0: 0 (16.0)
|   |   |   |   A9 = 1: 1 (16.0)
|   A2 = 1:
|   |   A3 = 1: 1 (64.0)
|   |   A3 = 0:
|   |   |   (ST3)
|   |   |   A7 = 0:
|   |   |   |   A9 = 0: 1 (16.0)
|   |   |   |   A9 = 1: 0 (16.0)
|   |   |   A7 = 1:
|   |   |   |   A8 = 0: 0 (16.0)
|   |   |   |   A8 = 1:
|   |   |   |   |   A9 = 0: 0 (8.0)
|   |   |   |   |   A9 = 1: 1 (8.0)
```

Function 9 dataset: The function involves 4 attributes:

(a) **Salary: uniformly distributed from 20000 to 50000.**

(b) **Commission: is 0 if Salary is greater than 75000, it is uniformly distributed from 10000 to 75000 otherwise.**

(c) **Elevel: discrete and uniformly chosen from** $[0, 1, 2, 3, 4]$**.**

(d) **Loan: uniformly distributed from 0 to 50000.**

The patterns are generated randomly according to the function definition:

Class 1 *0.67(Salary + Commission) - 5000 Elevel - 0.2 Loan > 10000*

Class 2 Otherwise.

C4.5 decision trees:

Training patterns	C4.5 Tree size	Classification Error
500	25	4.4 %
1000	57	2.9 %
2000	121	1.9 %

3. Neural networks are trained using all patterns.

Heart disease dataset:

Attribute	Type	Possible values/Range	Network inputs
$\mathcal{A}_1$	Continuous	$[29, 77]$	$\mathcal{I}_1$
$\mathcal{A}_2$	Discrete	0 or 1	$\mathcal{I}_2$
$\mathcal{A}_3$	Discrete	1,2,3 or 4	$\mathcal{I}_3, \mathcal{I}_4, \mathcal{I}_5, \mathcal{I}_6$
$\mathcal{A}_4$	Continuous	$[94, 200]$	$\mathcal{I}_7$
$\mathcal{A}_5$	Continuous	$[126, 564]$	$\mathcal{I}_8$
$\mathcal{A}_6$	Discrete	0 or 1	$\mathcal{I}_9$
$\mathcal{A}_7$	Discrete	0,1 or 2	$\mathcal{I}_{10}, \mathcal{I}_{11}, \mathcal{I}_{12}$
$\mathcal{A}_8$	Continuous	$[71, 202]$	$\mathcal{I}_{13}$
$\mathcal{A}_9$	Discrete	0 or 1	$\mathcal{I}_{14}$
$\mathcal{A}_{10}$	Continuous	$[0, 6.2]$	$\mathcal{I}_{15}$
$\mathcal{A}_{11}$	Discrete	1,2 or 3	$\mathcal{I}_{16}, \mathcal{I}_{17}, \mathcal{I}_{18}$
$\mathcal{A}_{12}$	Discrete	0,1,2 or 3	$\mathcal{I}_{19}, \mathcal{I}_{20}, \mathcal{I}_{21}, \mathcal{I}_{22}$
$\mathcal{A}_{13}$	Discrete	3,6 or 7	$\mathcal{I}_{23}, \mathcal{I}_{24}, \mathcal{I}_{25}$

Extracted rule:

If $-0.07\mathcal{A}_4 + 3.66\mathcal{A}_8 < -0.66$ and $1.99\mathcal{A}_{10} \geq -0.60$, then class 1,

else if $1.99\mathcal{A}_{10} > 0.25$, then class 1,

else class 0.

- NN rule accuracy: 88.38 % (training set), 84.85 % (testing set).
- C4.5 rule:
  - Rule accuracy: 92.9 % (training set), 74.7 % (testing set).
  - Number of rules: 11.

4. Neural networks can generate DNF, MofN or oblique rules.

   Splice junction dataset:

   - Each sample in the dataset is a 60-nucleotide-long DNA sequence.
   - A nucleotide may assume one of the 4 possible values: G, T, C, or A.
   - Training set: 1006 patterns, testing set: 3175 patterns.
   - 3 possible classes: IE, EI, N.
   - Extracted rule:

     If 3 of (@+1=G, @+2=T, @+3=A, @+5=G),
       then EI,
     else if 3 of (@-3=T, @-3=C, @-2=A, @-1=G),
       then IE,
     else N.
   - Testing set accuracy: 92.94 %
   - C4.5 accuracy: 92.13 %, the number of rules: 43.