Neural Network Structures with Constant Weights to Implement Dis-Jointly Removed Non-Convex (DJRNC) Decision Regions: Part B - Nested, and Disconnected Cases

CHE-CHERN LIN
National Kaohsiung Normal University
Department of Industrial Technology Education
116 Ho Ping First Road, Kaohsiung,
TAIWAN, R.O.C.

*Abstract:- I*n this paper, we propose two more complicated DJRNC decision regions: nested and disconnected decision regions. We first give the definitions of the decision regions and explain how to implement these regions using multi-layer perceptrons. We prove the feasibilities and discuss the conceptual structures of hardware implementation to realize these DJRNC decision regions.

Key-words: - Multi-layer perceptron; Nested region; Non-convex region, Disconnected region.

1. Nested DJRNC Decision Regions

Let D_0 be the input space, and D_1, D_2… ,D_p be a series of recursively included DJRNC polyhedrons in the input space with the following relation [1]:

$$D_0 \supset D_1 \supset D_2 \supset \cdots \supset D_p \quad (1)$$

A "nested DJRNC decision region" S is defined as follows [1]:

$$S=(D_0 \cap \overline{D}_1)\cup(D_2 \cap \overline{D}_3)\cup\cdots\cup(D_{p-1} \cap \overline{D}_p) \quad (2)$$

where $\overline{D}$ denotes the complement of D. Fig. 1(a) shows an example of a nested DJRNC decision region consisting of three recursively included polyhedrons: D_1, D_2, and D_3.

The neural network to solve the nested DJRNC decision regions is a four-layer perceptron. Fig. 2 is an example of the four-layer perceptron to implement the nested DJRNC decision region shown in 1(a). The first layer of the network serves to form the nested DJRNC decision region. The second layer determines whether a pattern is in the MCCP or an RCP associated with a particular DJRNC polyhedron or not. The third layer detects the DJRNC polyhedron in which the pattern resides. The fourth layer determines whether the pattern belongs to class A or class B. All of the weights in the second and fourth layers are set to be 1's. Each of the third layer weights is 1 if it is connected with a MCCP, and –1 if connected with an RCP, as mentioned in the previous section. The threshold for a second layer node (θ_h) is equal to the number of the bounding hyper-planes of its associated MCCP or RCP. The threshold for each node in the third layer (θ_h) is set to 1. The output layer (the fourth layer) consists of only one node. The activation function for the output node y is of the form:

$$y=\begin{cases}(v+1) \bmod 2, & \text{if } D_0 \cap \overline{D}_1 \text{ belongs to class A}\\ v \bmod 2, & \text{if } D_0 \cap \overline{D}_1 \text{ belongs to class B}\end{cases}$$

(3)

where notation 'mod' denotes a modulus (remainder) operation of two integer numbers.

The proof of implementation feasibility of the nested DJRNC decision regions by the proposed four-layer perceptron is straightforward. We explain it by Fig. 1 and Fig. 2. In Fig. 1, the nested DJRNC decision region is formed by three DJRNC polyhedrons including D_1, D_2, and D_3. In Fig. 2 $D_0 \cap \overline{D}_1$ belongs to class B. If a pattern is located in $D_0 \cap \overline{D}_1$, none of the three DJRNC polyhedrons includes the pattern. v is therefore equal to 0. By Eq. (3), y = 0 (class B). If a pattern is located in $D_1 \cap \overline{D}_2$, only D_1 includes the pattern. v is equal to 1, and y = 1 (class A). If a pattern is located in $D_2 \cap \overline{D}_3$, both D_1 and D_2 include the pattern. v is then equal to 2, and y = 0 (class B). If a pattern is located in D_3, all of the three DJRNC polyhedrons (D_1, D_2, and D_3) include the pattern. v is equal to 3, and y = 1 (class A).

Similarly, one can use the same procedure to sequentially obtain alternative outputs (0 and 1) for any nested DJRNC decision regions.

Using the same procedure as above, one can easily prove the feasibility for the case of $D_0 \cap \overline{D}_1$ belonging to class A. Fig. 2 is the four-layer perceptron to implement the nested DJRNC decision region shown in Fig. 1(a).

2. Disconnected DJRNC Regions

A "disconnected DJRNC decision region" is a region containing two or more individual DJRNC polyhedrons. Fig. 3(a) is an example of a disconnected DJRNC decision region containing two DJRNC polyhedrons: DJRNC polyhedron1 and DJRNC polyhedron 2. The disconnected DJRNC decision region can be solved using a four-layer perceptron. The first three layers of the four-layer perceptron serve to implement the individual DJRNC polyhedrons in a disconnected DJRNC decision region. The fourth layer of the four-layer perceptron performs a logic "OR" operation and determines if an input pattern is in "any" of the DJRNC polyhedrons in the disconnected DJRNC decision region [2]. The weights and the thresholds of the hard limiters of the first three layers of the four-layer perceptron are the same as those of the three-layer perceptron implementing single DJRNC decision regions. The weights linking the third layer nodes and the fourth layer node (the output node) are 1's. The threshold of the hard limiter for the fourth layer node is 1. Fig. 4 is the four-layer perceptron to implement the disconnected DJRNC decision region shown in Fig. 3(a).

3. Discussion

In the proposed network structures, the weights and parameters of the activation functions are pre-determined when the DJRNC decision regions are established. Each weight determined by this paper is either 1 or –1. Therefore, it is easy to implement these decision regions by hardware implementations. The conceptual structure of hardware implementation to solve the nested DJRNC decision regions is a four-layered network. In this structure, the first layer serves to establish the nested DJRNC decision regions and we need multipliers and adders to form the decision regions. The second layer determines whether a pattern reside in a particular RCP or MCCP. We only need adders and comparators to perform the functionality of the second layer, since the weights of the second layer are all 1's. The third layer detects the DJRNC polyhedron in which the pattern resides. Each of the third layer weights is either 1 or –1. We then use multipliers (multiplying by 1 or –1), adders and comparators to perform the functionality of the third layer. The activation function of the fourth layer is the modulus operation dividing the input of the activation function by 2. The output of the activation function will be sequentially alternative (0 or 1) when the integer value of the input of the activation function increases (0, 1, 2, 3, 4, …). This is realized by taking the least significant bit of the input of the activation function. Therefore we use an adder and take its least significant bit to perform the functionality of the fourth layer.

Similarly, one can use the same idea to get the hardware implementations of solving the disconnected DJRNC decision regions.

4. Conclusions

We proposed neural network models to implement three different types of DJRNC decision regions including single, nested, and disconnected DJRNC regions. In the proposed models, no constructive algorithm is needed for implementing the above three decision regions. Each weight determined by this paper is either 1 or –1. We defined single, nested, and disconnected DJRNC decision regions and presented three implementing multi-layer perceptrons to solve the three DJRNC decision regions. We also proved the implementation feasibilities and discussed the issues of hardware implementations for the proposed multi-layer perceptrons.

As for the direction of further studies, we suggest to generalize the models to solve more complicated decision regions and use electrical circuits to realize the proposed neural network models.

References

[1] C. Cabrelli, U. Molter, R. Shonkwiler, A Constructive Algorithm to Solve Convex Recursive Deletion (CoRD) Classification Problems via Two-layer Perceptron Networks, *IEEE Trans. On Neural Networks*, Vol. 11, No. 3, 2000, pp. 811-816.

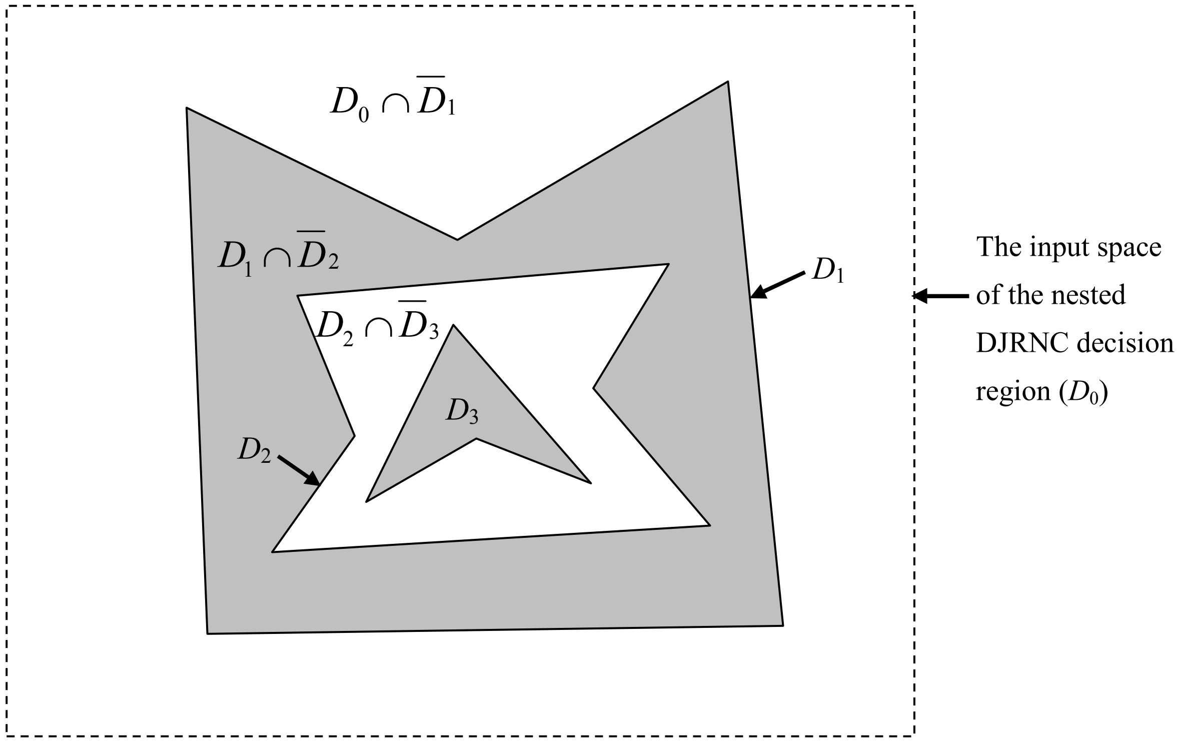
[2] S. Draghici, The Constraint Based Decomposition (CBD) Training Architecture, *Neural Networks* 14, 2001, pp. 527-550.

[3] C. Lin, Partitioning Capabilities of Multi-layer Perceptrons on Nested Rectangular Decision Regions Part I: Algorithm, *WSEAS Transactions on Information Science and Applications*, Issue 9, Volume 3, September, 2006, pp. 1674-1680.

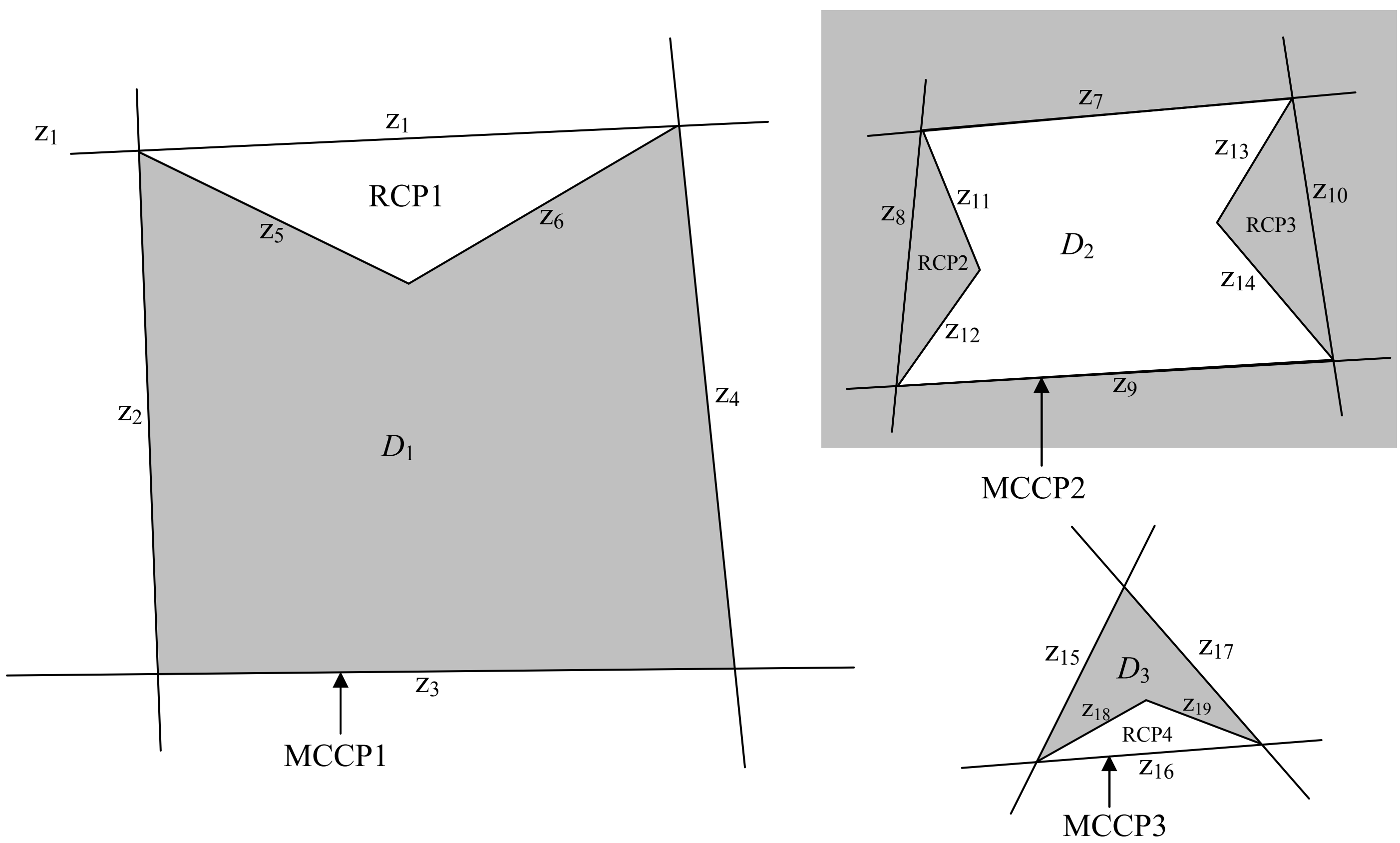
[4] C. Lin, Partitioning Capabilities of Multi-layer Perceptrons on Nested Rectangular Decision Regions Part II: Properties and Feasibility, *WSEAS Transactions on Information Science and Applications*, Issue 9, Volume 3, September, 2006, pp. 1681-1687.

[5] C. Lin, Partitioning Capabilities of Multi-layer Perceptrons on Nested Rectangular Decision Regions Part III: Applications, *WSEAS Trans. on Information Science and Applications*, Issue. 9, Volume. 3, September, 2006, pp. 1688-1694.

[6] C. Lin, A Constructive Algorithm to Implement Celled Decision Regions Using Two-Layer Perceptrons, *WSEAS Transactions on Information Science and Applications*, Issue. 9, Volume 3, September, 2006, pp. 1654-1660.



(a) The nested DJRNC decision region



(b) The three DJRNC polyhedrons (D_1, D_2, and D_3) and their bounding hyper-planes.

Fig. 1: An illustrative example of a nested DJRNC decision region.

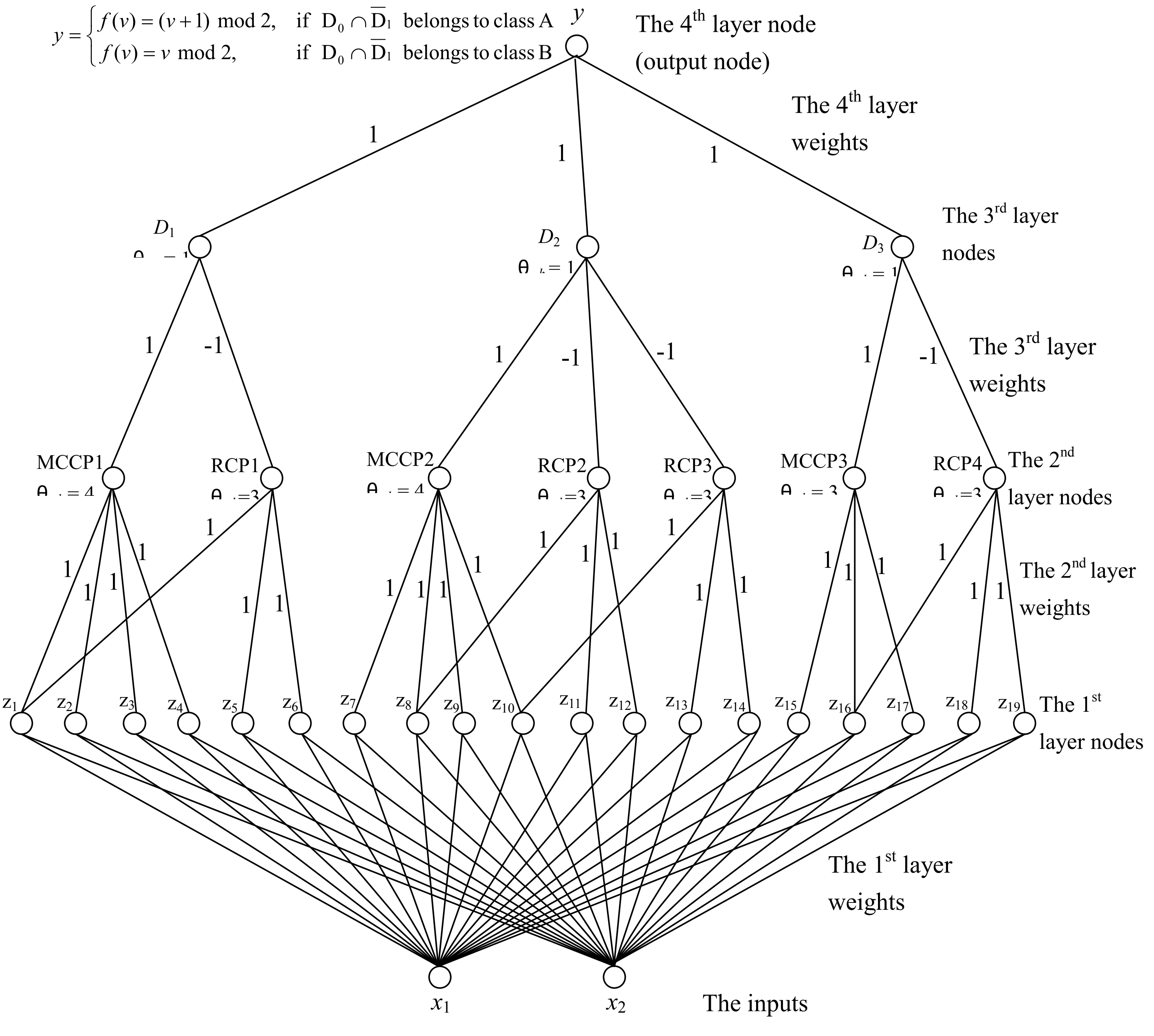
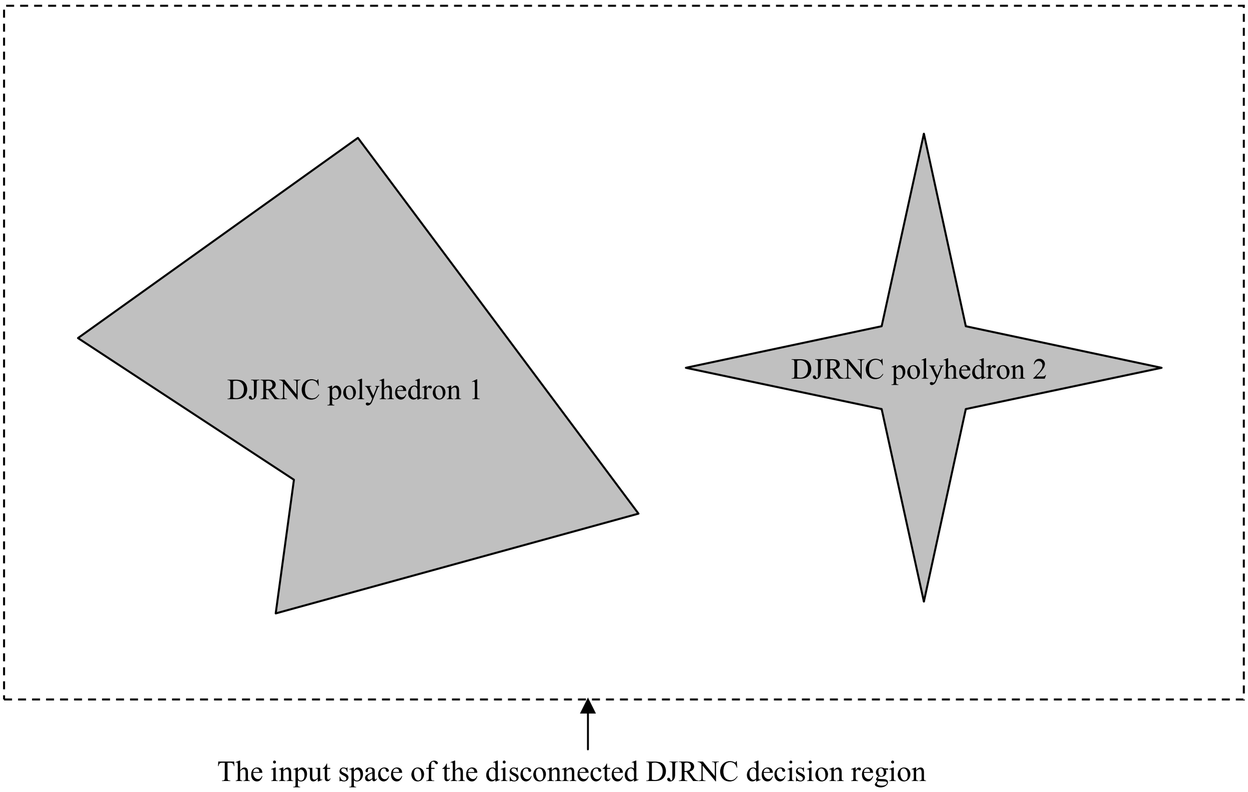
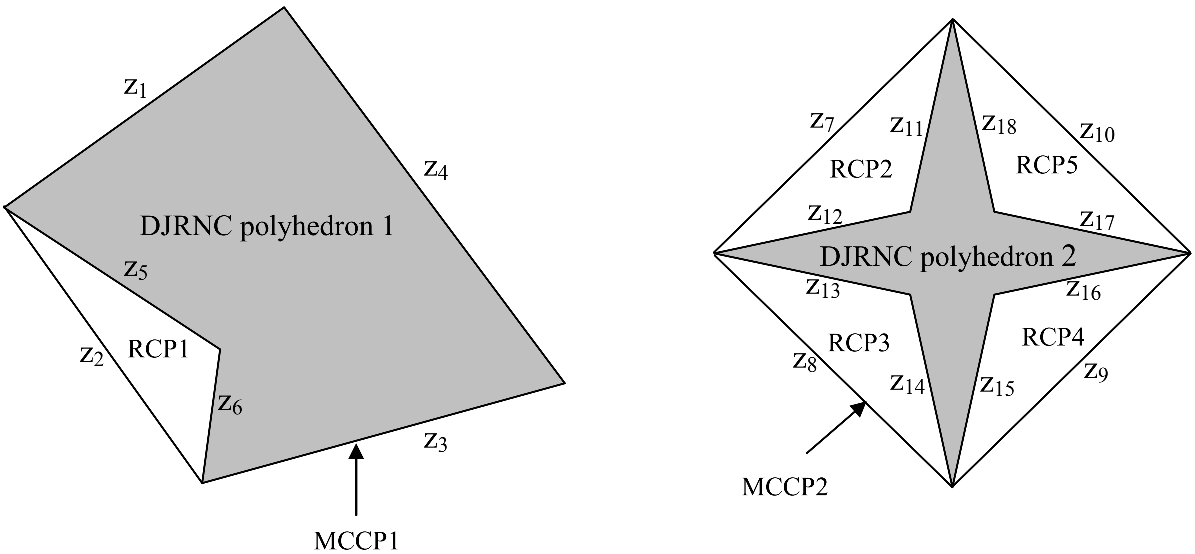


Fig. 2: The four-layer perceptron to implement the nested DJRNC decision region shown in Fig. 1(a).



(a) The disconnected DJRNC decision region.



(b) The MCCPs, RCPs and the bounding hyper-planes of the disconnected DJRNC decision region.

Fig. 3: An illustrative example of a disconnected DJRNC decision region.

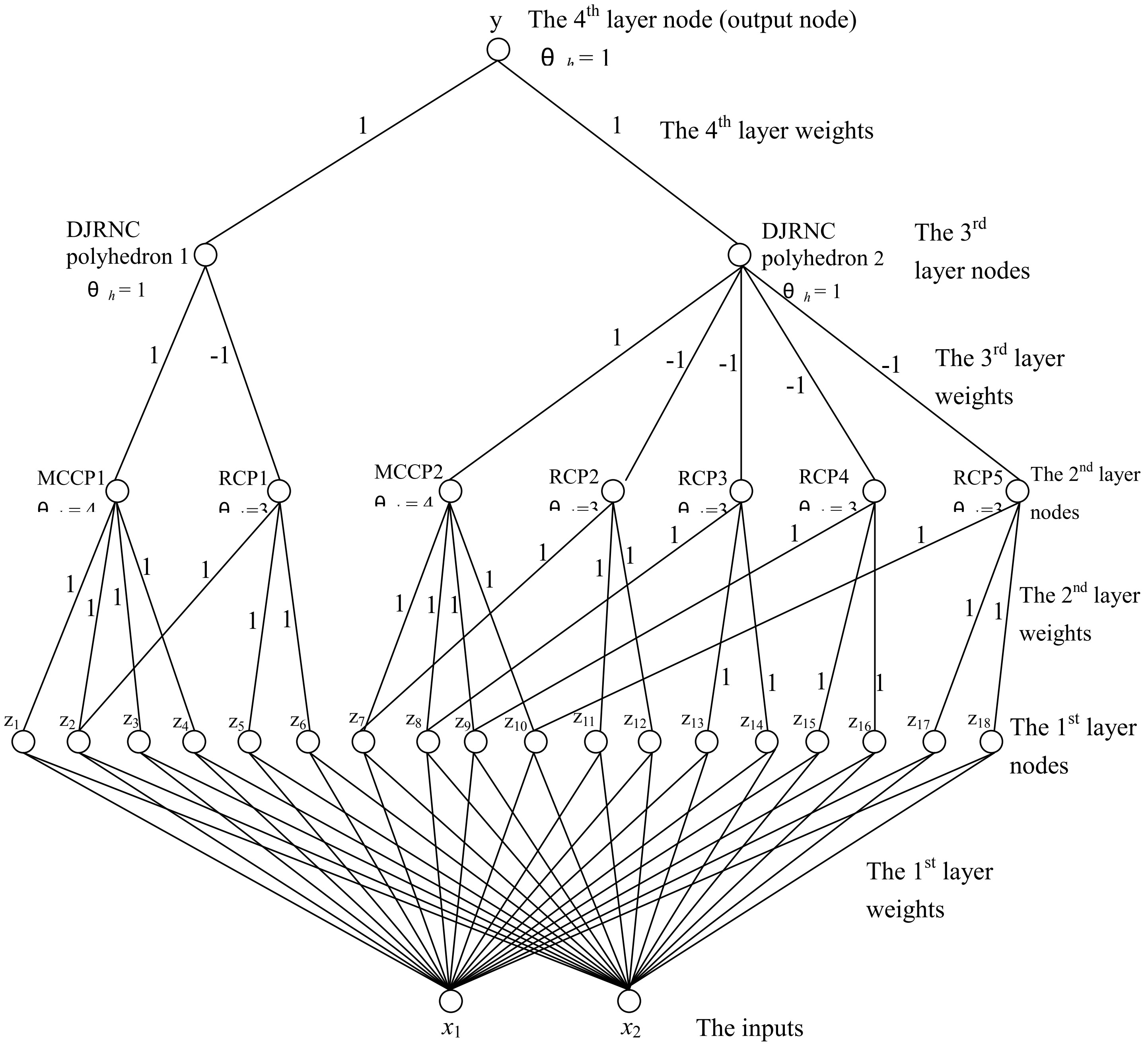


Fig. 4: The four-layer perceptron to implement the disconnected DJRNC decision region shown in Fig. 3(a).