Neural Network Structures with Constant Weights to Implement Dis-Jointly Removed Non-Convex (DJRNC) Decision Regions: Part A - Properties, Model, and Simple Case

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

Abstract - This is a series of studies to discuss the partitioning capabilities of multi-layer perceptrons on dis-jointly removed non-convex (DJRNC) decision regions. There are two papers proposed in the series of studies including part A and part B. In part A, we propose a network structure to implement DJRNC decision regions using multi-layer perceptrons. In the proposed structure, all weights and the parameters of the activation functions are pre-determined when a DJRNC decision region is established. No constructive algorithm is needed for implementing the DJRNC decision regions and each weight determined by this paper is either 1 or –1. This makes the hardware implementations of the proposed network structures easy. Three cases are discussed in this paper including single, nested, and disconnected decision regions. The first case is shown in part A, and the rest of two are demonstrated in part B. We also provide three multi-layer perceptrons to implement the three decision regions and prove the implementation feasibilities of the proposed model

Key-Words: - Multi-layer perceptron; Nested decision region; Non-convex decision region; Disconnected decision region.

1. Introduction

A multi-perceptron is one of popular neural network structures for implementing classification problems. Adjustable weights are used to connect the nodes between adjacent layers and optimized by training algorithms to get the desired classification results. However, using training algorithm to optimize the weights probably needs thousands of iterations and hence spends a lot of computational time. Related studies about partitioning capabilities of multi-layer perceptrons on vertical and horizontal partitioning have been discussed in [1-4]. The rest of studies on the partitioning capabilities of multi-layer perceptrons, see [5-13].

The implementation feasibility of convex recursive deletion regions has been discussed in [9] where the authors proposed a constructive algorithm to determine the parameters (weights and thresholds) of two layer perceptrons.to implement the convex recursive deletion decision regions. Non-convex decision regions are more complicated case. In this paper, the author presents a network structure to implement **Dis-Jointly Removed Non-Convex (DJRNC)** decision regions using multi-layer perceptrons. In the proposed structure, all weights and the parameters of the activation functions are pre-determined when a DJRNC decision region is established. No constructive algorithm is needed for implementing the DJRNC decision regions and each weight determined by this paper is either 1 or –1.

In the following, we present an illustrative example to explain how a two-layer perceptron implements a convex decision region. For the visual reason, in this paper we use two-dimensional examples. It is easy to generalize them to multi-dimensional cases.

Fig. 1(a) is a convex decision region containing a convex polyhedron (C) which is bounded by 5 hyper-planes: z_1, z_2, z_3, z_4, and z_5. In this figure, the dotted rectangular box indicates the input space of the convex decision region where the shaded area belongs to class A while the blank area belongs to class B. We call the hyper-planes bounding the convex polyhedron ***"bounding hyper-planes"***. A bounding hyper-plane divides the input space into two linearly separable half hyper-planes. The '1' side of a bounding hyper-plane is the half hyper-plane containing the convex polyhedron, and '0' side is the other half hyper-plane, as indicated in Fig. 1(a).

In a convex decision region, a pattern is said to be in the convex polyhedron if and only if it is on the '1' sides of all bounding hyper-planes of the convex polyhedron. It has been known that a convex decision region can be implemented by a two-layer perceptron where the first layer forms the convex decision region and the second layer performs the classifications. The first layer weights are pre-determined when the decision region is formed. The second layer weights are all 1's. We use a hard limiter as an activation function in the second layer node to get the desired classification results. The hard limiter is of the form [1, 8]

$$y = g(\theta) = \begin{cases} 1 & \text{if} \quad \theta \geq \theta_h \\ 0 & \text{if} \quad \theta < \theta_h \end{cases} \tag{1}$$

where θ_h is the threshold and set to be equal to the number of the bounding hyper-planes of the convex polyhedron. For example, the convex decision region shown in Fig. 1(a) can be implemented by the two-layer perceptron shown in Fig. 1(b) where the threshold (θ_h) of the hard limiter in the second layer is set to 5 since the convex polyhedron is bounded by 5 bounding hyper-planes (z_1 to z_5).

Nested convex decision region (or convex recursive deletion region) problems have been solved by a constructive algorithm via two-layer perceptrons [5].

Next, we give some definitions necessary for the paper.

A DJRNC polyhedron D is a polyhedron obtained by dis-jointly removing a series of convex polyhedrons (C_1, C_2, …, C_n) from an original convex polyhedron (C_0), given by

$$D = C_0 - C_1 - C_2 - \cdots - C_n \tag{2}$$

where C_0 (the original convex polyhedron) is called the ***"minimum containing convex polyhedron (MCCP)"*** of D; C_0, C_1, …, and C_n are called the "***removed convex polyhedrons (RCPs)***" of D.

The word "*dis-jointly*" means that any two of the RCPs do not have any common bounding hyper-planes except for the bounding hyper-planes of the MCCP.

Fig. 2(a) shows a DJRNC polyhedron (D) in a single DJRNC decision region (defined in the next section). Fig. 2(b) shows the MCCP and the three RCPs of the DJRNC polyhedron (RCP1, RCP2, and RCP3). Fig. 2(c) demonstrates the bounding hyper-planes of the MCCP and the three RCPs. The '0' sides and '1' sides of these bounding hyper-planes are also displayed in Fig. 2(c).

2. Model and Feasibility

2.1 Single DJRNC decision region Model

A "*single DJRNC decision region*" is a region containing only one DJRNC polyhedron. Fig. 2 (a) is also an example of a single DJRNC decision region.

A single DJRNC decision region can be implemented by a three-layer perceptron where all of the parameters (weights and thresholds) are constants. In the three-layer perceptron, the first layer serves to form the decision region, the second layer detects whether a pattern resides in the MCCP or a particular RCP, and the third layer makes the final classification. Each node in the second layer is associated with the MCCP or a particular RCP with one-to-one correspondence. We use the same notation to present a node of the second layer and its corresponding MCCP or RCP. The weights of the second layer are all 1's. Each of the third layer weights is determined as follows: if it is connecting the MCCP node with the third layer node, the weights is set to 1; if it is connecting any of the RCPs nodes with the third layer node, the weight is set to –1. The threshold for a particular node of the second layer is equal to the number of the bounding hyper-planes of the MCCP or an RCP associated with the particular node. The threshold (θ_h) for the third layer node (the output node) is 1.

2.2 Feasibility

We prove the implementation feasibility by considering the following cases:

(1) If the pattern doesn't reside in the MCCP, the final classification result is 0 (class B).

(2) If the pattern resides in the DJRNC polyhedron, only the MCCP node produces a '1'. The final classification result is 1 (class A).

(3) If the pattern resides in a particular RCP, both the MCCP and the particular RCP produce '1's. The weight connecting the MCCP node with the third layer node is 1, and the weight connecting the particular RCP with the third layer node is –1. The sum of them is 0. Therefore the final classification is 0 (class B).

Fig. 3 is the three-layer perceptron to implement the single DJRNC decision region shown in Fig. 2 (a).

3. Conclusions

We proposed a neural network model to solve the DJRNC decision regions using multi-layer perceptrons. In the proposed multi-layer perceptrons, all weights and the parameters of the activation functions are pre-determined when a DJRNC decision region is established. No constructive algorithm is needed for implementing the above three decision regions. Each weight determined by this paper is either 1 or –1. We presented three cases to discuss, in practical, the capabilities of multi-layer perceptrons on DJRNC decision regions including single, nested, and disconnected DJRNC decision regions.

There are two papers proposed in the series of studies including part A and part B. In this paper (part A), we defined single DJRNC decision regions and presented the implementing multi-layer perceptrons for the DJRNC decision regions. We also proved the implementation feasibility for the single DJRNC decision region.

References

[1] 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.

[2] 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.

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

[4] 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.

[5] J. Makhoul, A. El-Jaroudi, R. Schwartz, Partitioning Capabilities of Two-layer Neural Networks, *IEEE Trans. on Signal Processing*, Vol. 39, No. 6, 1991, pp. 1436-1440.

[6] R. P. Lippmann, An Introduction to Computing with Neural Nets, *IEEE ASSP Mag*., Vol.4, 1987, pp. 4-22.

[7] R. Shonkwiler, Separating the Vertices of N-cubes by Hyperplanes and its Application to Artificial Neural Networks, *IEEE Trans. on Neural Networks*, Vol. 4, No. 2, 1993, pp. 343-347.

[8] C. Lin, A. El-Jaroudi, An Algorithm to Determine the Feasibilities and Weights of Two-Layer Perceptrons for Partitioning and Classification, *Pattern Recognition*, Vol. 31, No. 11, 1998, pp. 1613-1625.

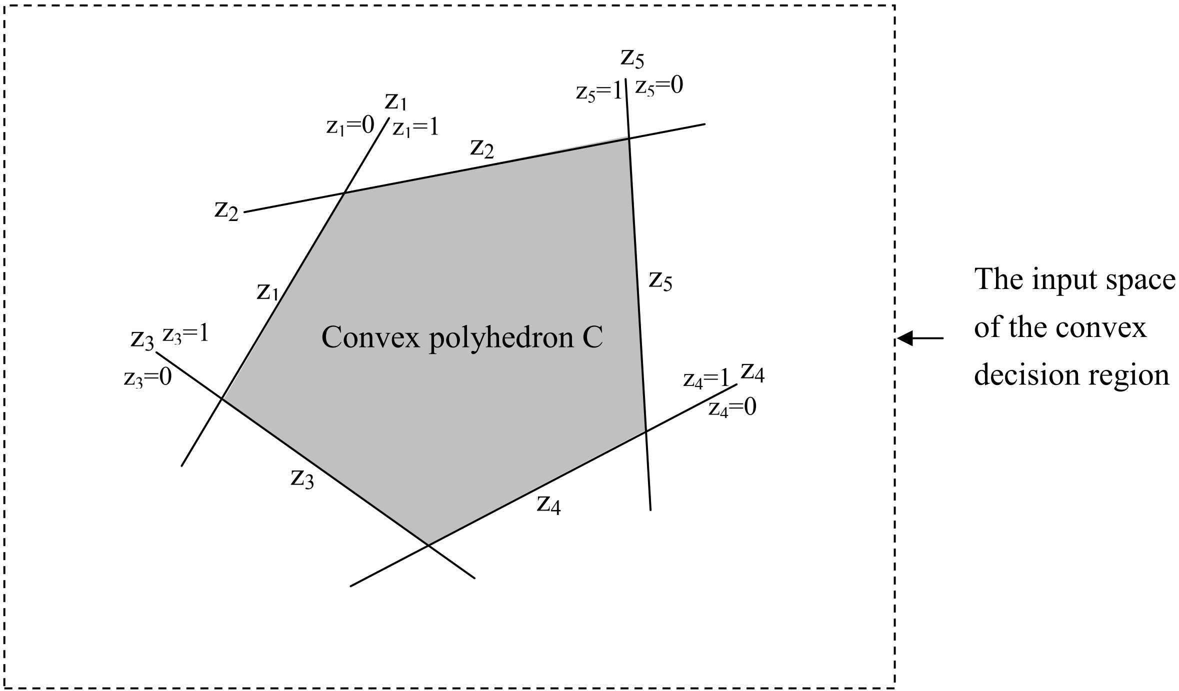
[9] C. Cabrelli, U. Molter, R. Shonkwiler, A Constructive Algorithm to Solve Convex recursive deletion (CoRD) Classification Problems via Two-layer Perceptron Networks, *IEEE Trans. On Neural Networks*, Vol. 11, No 3, 2000, pp. 811-816.

[10] W. Fan, L Zhang, Applying SP-MLP to Complex Classification Problems, *Pattern Recognition Letters*, 21, 2000, pp. 9-19.

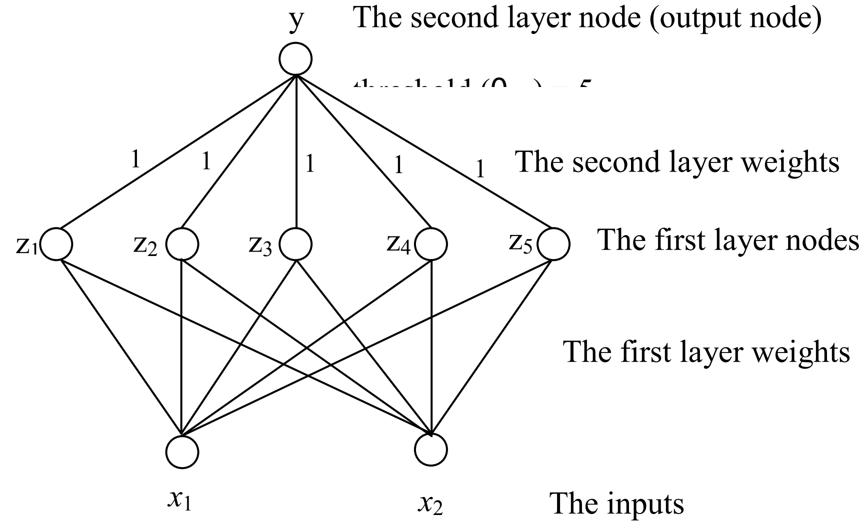
[11] V. Deolalikar, A Two-layer Paradigm Capable of Forming Arbitrary Decision Regions in Input Space, *IEEE Trans. On Neural Networks*, Vol. 13, No. 1, 2002, pp. 15-21.

[12] G. Huang, Y Chen, H. A. Babri, Classification Ability of Single Hidden Layer Feedforward Neural Networks, *IEEE Trans. On Neural Networks*, Vol. 11, No. 3, 2000, pp. 799-801.

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

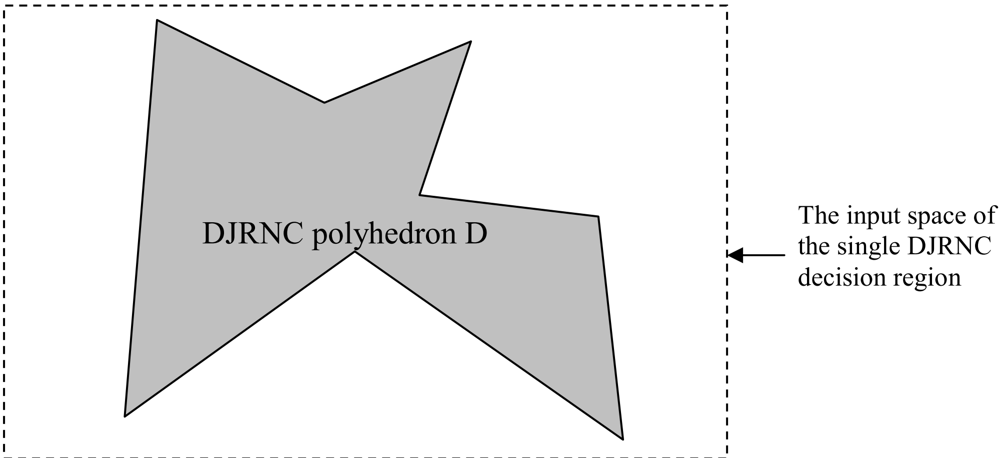


(a) The convex decision region.

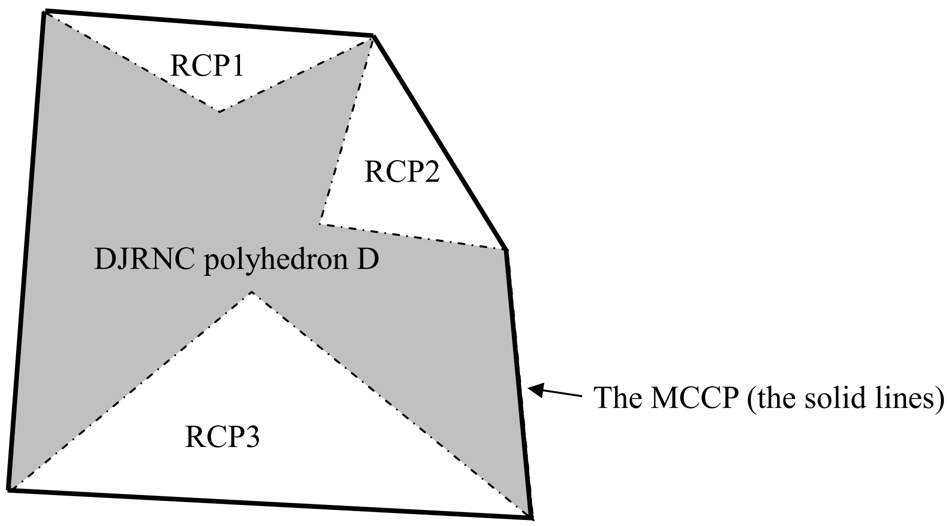


(b) The two-layer perceptron to implement the convex decision region shown in (a).

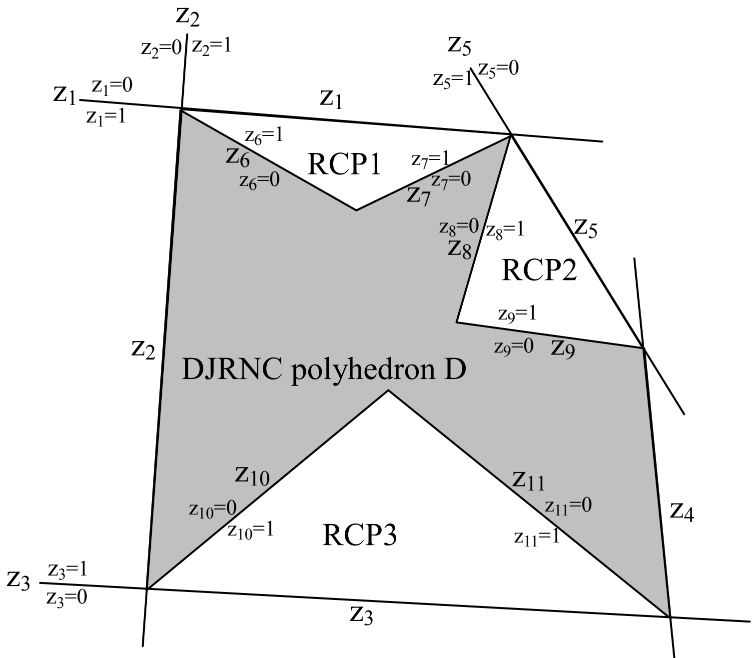
Fig. 1: An illustrative example of a convex decision region implemented by a two-layer perceptron.



(a) The single DJRNC decision region



(b) The MCCP and the three RCPs (RCP1, RCP2, and RCP3) of DJRNC polyhedron D.



(c) The bounding hyper-planes

Fig. 2: An illustrative example of a single DJRNC decision region

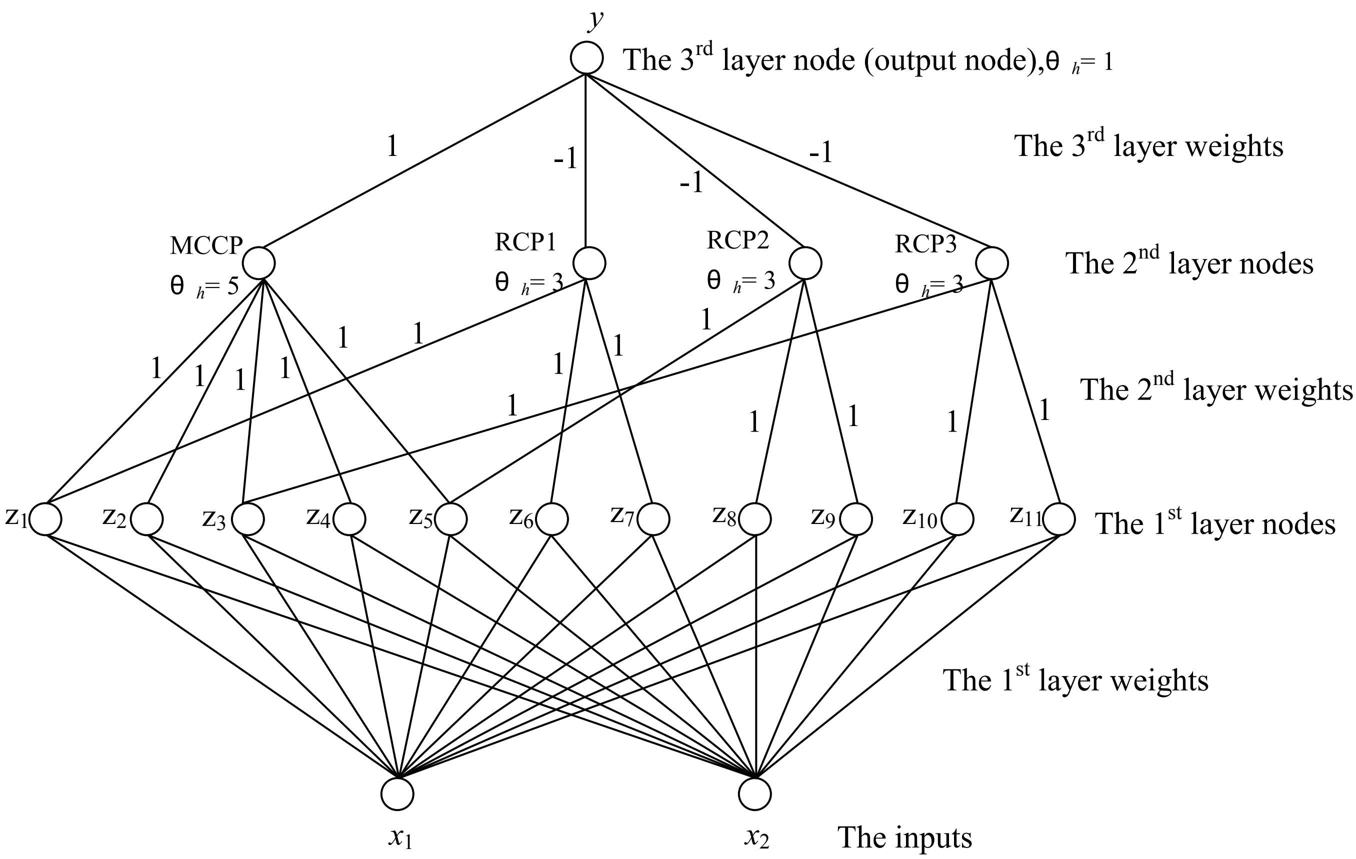


Fig. 3: The three-layer perceptron to implement the single DJRNC decision region shown in Fig. 2(a).