Proceedings of the 7th WSEAS International Conference on Neural Networks, Cavtat, Croatia, June 12-14, 2006 (pp126-131)

A Constructive Algorithm to Determine the Feasibility and Weights of Two-Layer Perceptrons for Celled Decision Regions

CHE-CHERN LIN
Department of Industrial Technology Education
National Kaohsiung Normal University
116 Ho Ping First Road,Kaohsiung, 802
TAIWAN, ROC

Abstract: - Necessary and sufficient conditions for implementing particular decision regions by multi-layer perceptrons have been presented in recent studies. In this paper, from a viewpoint of engineering, a constructive algorithm is proposed to implement celled decision regions using two-layer perceptrons without any training procedure. The algorithm examines the feasibility of a celled decision region and then determines the weights of the second layer for a particular two-layer perceptron to implement the decision region if it is realizable. The algorithm is fast based on two aspects. First, it tests the feasibility and determines the weights without any training procedure. Second, the classifications of input patterns are based on integer manipulations since the weights determined by the algorithm are all integers. The proposed algorithm consists of only three simple steps and is implemented easily by computer programming languages.

Key-Words: - Classification; Multi-layer perceptron; Celled decision region; Partitioning capability.

1 Introduction

A multi-layer perceptron is a layered structure neural network. People use weights to connect nodes in each layer to perform the desired mapping from inputs to outputs. Training algorithms are used to adjust the weight in a neural network in order to get better classification results. However, it takes a lot of time to optimize the weights during the training procedure especially when the number of weights in a neural network is large.

Another approach to obtain the weights of a neural network is to examine its partitioning capability and then determine the weights using some appropriate algorithms. An original discussion about the partitioning capabilities of multi-layer perceptrons can be found in [1] where the author indicated one-layer perceptrons only realize linearly separating decision regions, two-layer percptrons (TLPs) implement either convex open or closed decision regions, and three-layer perceptrons can successfully partition arbitrary decision regions. Further studies related to the feasibilities of multi-layer perceptrons are listed in [2-11]. The partitioning capability of TLPs has been generally discussed in [8], in which the author proposed the Weight Deletion/Selection Algorithm to determine the feasibility of TLPs and select the weights without any training procedure if a decision region is realizable.

In this paper, the author proposes a constructive algorithm to examine the feasibility of a celled decision region by a particular TLP, and then determine the weights of second layer of the TLP if the decision region is realizable. Therefore, in this paper, when the author mentions **‘***select weights* **’**, it means *‘select weights of the second layer of a particular TLP* **’**.

The primary advantage of the algorithm lies in the computational speed to determine the feasibility and weights of the neural networks for celled decision regions because there is no training procedure to be used to adjust the weights. Furthermore, the weights determined by the algorithm are all integers. This will make the computation faster when classifying input patterns.

2 Preliminaries

2.1 Decision regions

It has been known that the weights of the first layer of a TLP are pre-determined if its associated decision region is established [2, 8]. We only need to get the weights of the second layer of the TLP. This paper focuses on how to generate the weights of the second layer of a TLP and assumes the weights of the first layer of a TLP are pre-determined.

It has been known that for a two-layer perceptron, the number of inputs determines the dimensionality of its associated decision region. The first layer of a two-layer perceptron generates the associated decision region and the second-layer serves to map the input pattern to the desired outputs [2,8]. An m-dimensional decision region can be divided by a group of m-dimensional partitioning hyper-planes. These partitioning hyper-planes are associated with the hidden nodes of a particular two-layer perceptron with a one-to-one correspondence and divide the original decision region into different sub-regions.

For two-class classification problems, the θ value for a particular sub-region in a decision region, is defined as follows [8]

$$\theta_l = \sum_{k=1}^{p} w_k z_k \tag{1}$$

where θ_l is the θ value of sub-region l, p is the number of partitioning lines in the decision region, and w_k is the weight connecting first layer node z_k with the output node.

The output of the TLP is given by

$$y = \begin{cases} 1 \quad (\text{class } A) \quad \text{if } \theta_l \geq \theta_h \\ 0 \quad (\text{class } B) \quad \text{if } \theta_l < \theta_h \end{cases} \tag{2}$$

where θ_h is the threshold for the output node.

The necessary and sufficient condition for implementing a decision region is the minimum θ value of sub-regions belonging to class A must be greater than the maximum θ value of sub-regions belonging to class B [2,6,8].

2.2 Celled decision regions

A ***'celled decision region'*** is a decision region partitioned by horizontal and vertical lines by which the decision region is divided into rectangle-like celled sub-regions. These celled sub-regions are then grouped into several horizontal or vertical strips. If a celled decision has m vertical partitioning lines and n horizontal partitioning lines, we get (m+1) vertical strips and (n+1) horizontal strips. For convenience, the author uses notation **'VS_i'** to represent vertical strips i and numbers the vertical strips from the left to the right (VS_0, VS_1, VS_2, …, VS_m). Similarly, the author uses notation **'HS_j'** to represent horizontal strip j and numbers the horizontal strips from the bottom to the top (HS_0, HS_1, HS_2, …, HS_n.). A celled sub-region is denoted as 'C_{ij}' if it is in the intersection of vertical strip i (VS_i) and horizontal strip j (HS_j). Figure 1 displays a celled decision region with 4 horizontal partitioning lines (w_{h1} to w_{h4}) and 5 vertical partitioning lines (w_{v1} to w_{v5}). The decision region is established either by 6 vertical strips: VS_0, VS_1, VS_2, VS_3, VS_4 and VS_5 or by 5 horizontal strips: HS_0, HS_1, HS_2, HS_3, and HS_4. Each of vertical and horizontal strips consists of a series of celled sub-regions, as indicated in Figure 1.

A celled sub-region C_{ij} is represented by two components: ***'horizontal component'*** i and ***'vertical component'*** j. The author uses notation **'HC_i'** to represent horizontal component i and notation **'VC_j'** to represent vertical component j. It is very important to note that the horizontal component of C_{ij} (HC_i) is associated with vertical strip i (VS_i), and the vertical component of C_{ij} (VC_j) is associated with horizontal strip j (HS_j). For example, C_{23} is represented by HC_2 which is associated with vertical strip 2 (VS_2) and by VC_3 which is associated with horizontal strip 3 (HS_3). Furthermore, the author uses notation **'VS_{iA}'** to represent a subset of class A of VS_i, in which all elements belong to class A. The author also uses notation **'HS_{jA}'** to represent a subset of class A of HS_j, in which all elements belong to class A. For example, in Figure 1, $VS_{0A} = \{C_{01}, C_{02}, C_{03}, C_{04}\}$, $VS_{4A} = \{C_{41}, C_{42}\}$, $HS_{0A} = \Phi$ (the empty set), and $HS_{3A} = \{C_{03}, C_{23}\}$.

The author uses notation 'VS_{iA}^{V}' to represent a collection of the vertical components of the elements in VS_{iA},. For example, in Figure 1, $VS_{0A}^{V} = \{VC_1, VC_2, VC_3, VC_4\}$. Similarly, the author uses 'HS_{jA}^{H}' to represent a collection of the horizontal components of the elements in HS_{jA}. For example, in Figure 1, $HS_{2A}^{H} = \{HC_0, HC_1, HC_3, HC_4, HC_5\}$.

2.3 The XOR examining procedure

Based on the XOR criterion presented in [1-5], the author demonstrates the XOR examining procedure, by providing an example, to explain the necessary and sufficient condition of implementation of a celled decision region using a TLP. Observing Figure 1, one finds that in VS_2, C_{23} belongs to class A while C_{21} belongs to class B. To implement the classification, the order of the θ values of C_{23} and C_{21} is given by

$$\begin{aligned} \theta_{23} > \theta_{21} \Rightarrow w_{v1} + w_{v2} + w_{h1} + w_{h2} + w_{h3} > \\ w_{v1} + w_{v2} + w_{h1} \Rightarrow w_{h2} + w_{h3} > 0 \end{aligned} \tag{3}$$

Similarly, in VS_4, C_{41} belongs to class A while C_{43} belongs to class B. the order of the θ values of C_{41} and C_{43} is given by

$$\begin{aligned} \theta_{43} < \theta_{41} \Rightarrow w_{v1} + w_{v2} + w_{v3} + w_{v4} + w_{h1} + w_{h2} + w_{h3} < \\ w_{v1} + w_{v2} + w_{v3} + w_{v4} + w_{h1} \Rightarrow w_{h2} + w_{h3} < 0 \end{aligned} \tag{4}$$

Eqs. (3) and (4) lead to a contradiction, called the XOR problem [1-5], which means the decision region cannot be implemented by TLPs since the sum of w_{h2} and w_{h3} cannot be positive and negative values simultaneously.

However for a celled decision region, using the above XOR examining procedure to test the feasibilities spends a lot of computational time since it needs to examine any possible sub-region pairs where the XOR problem could occur. For example, consider a celled decision region with m vertical strips and n horizontal strips. There are $\frac{(m!)(n!)}{4}$ XOR pairs needed to be examined in the celled decision region (symbol '!' denotes a factorial manipulation). The computational complexity is extremely heavy when m and n are large numbers.

In Figure 1, VS_{2A}^V ={VC_3, VC_4}, and VS_{4A}^V ={VC_1, VC_2}. The XOR problem occurs because at least one element in VS_{4A}^V is not in VS_{2A}^V, and vice versa. For example, VC_1 is in VS_{4A}^V but not in VS_{2A}^V, and VC_3 is in VS_{2A}^V but not in VS_{4A}^V. To avoid the XOR problem in a celled decision region, for any possible pair of vertical strips, say VS_k and VS_l, the following relationship must be satisfied:

$$VS_{KA}^V \subseteq VS_{lA}^V \quad \text{or} \quad VS_{lA}^V \subseteq VS_{KA}^V \qquad (5)$$

Based on the above discussion, in the next section, two criteria are presented to test the feasibility of implementation of a celled decision region by a TLP and to select the weights of the second layer of the TLP if the decision region is realizable.

2.4 Criteria

Criterion 1: A celled decision region can be implemented by a TLP if for the vertical strips in the decision region one can find a particular order to satisfy the following relationship

$$\underbrace{VS_{iA}^V}_{\text{Position 1}} \subseteq \underbrace{VS_{jA}^V}_{\text{Position 2}} \subseteq \Lambda \quad \text{for distinct vertical strips} \qquad (6)$$

It is important to note that in Eq. (6) numbers i, j, k,…, etc, are not necessary in a numerical order. The author defines the '***position***' of a set in Eq. (6) according to the position counted from the left to the right. The '***rank***' of a vertical component VC_i, denoted as '***rank***(VC_i)', is said to be r if VC_i is in the set with position r but not in the set with position r-1. If a vertical component is in the set with posoition1, its rank is 1. For example, in Figure 2(a), we get VS_{0A}^V ={VC_0, VC_1, VC_2, VC_3, VC_4}, VS_{1A}^V ={VC_1, VC_2}, VS_{2A}^V ={VC_1, VC_2, VC_3, VC_4}, VS_{3A}^V ={VC_2}, and VS_{4A}^V ={VC_1, VC_2, VC_3}. Rearranging these sets, one gets the following relationship

$$\underbrace{VS_{3A}^V}_{\text{Position 1}} \subset \underbrace{VS_{1A}^V}_{\text{Position 2}} \subset \underbrace{VS_{4A}^V}_{\text{Position 3}} \subset \underbrace{VS_{2A}^V}_{\text{Position 4}} \subset \underbrace{VS_{0A}^V}_{\text{Position 5}} \qquad (7)$$

The ranks of the vertical components are as follows: ***rank***(VC_2) = 1, since VC_2 is in VS_{3A}^V (position 1), ***rank***(VC_1) = 2, since VC_1 is in VS_{1A}^V (position 2) but not in VS_{3A}^V (position 1),***rank***(VC_3) = 3, since VC_3 is in VS_{4A}^V (position 3) but not in VS_{1A}^V (position 2),***rank***(VC_4) = 4, since VC_4 is in VS_{2A}^V (position 4) but not in VS_{4A}^V (position 3), and ***rank***(VC_0) = 5, since VC_0 is in VS_{0A}^V (position 5) but not in VS_{2A}^V (position 4).

Similarly, to obtain the ranks of the horizontal components, one first gets the following relationship:

$$\underbrace{HS_{iA}^H}_{\text{Position 1}} \subseteq \underbrace{HS_{jA}^H}_{\text{Position 2}} \subseteq \Lambda \quad \text{for distinct horizontal strips} \qquad (8)$$

Again, in Figure 2(a), we get HS_{0A}^H ={HC_0}, HS_{1A}^H ={HC_0, HC_1, HC_2, HC_4}, HS_{2A}^H ={HC_0, HC_1, HC_2, HC_3, HC_4}, HS_{3A}^H ={HC_0, HC_2, HC_4}, and HS_{4A}^H ={HC_0, HC_2}.

Rearranging these sets, one gets the following relationship

$$\underbrace{HS_{0A}^H}_{\text{Position 1}} \subset \underbrace{HS_{4A}^H}_{\text{Position 2}} \subset \underbrace{HS_{3A}^H}_{\text{Position 3}} \subset \underbrace{HS_{1A}^H}_{\text{Position 4}} \subset \underbrace{HS_{2A}^H}_{\text{Position 5}} \qquad (9)$$

The ranks of the horizontal components are as follows: ***rank***(HC_0)=1, ***rank***(HC_2)=2, ***rank***(HC_4)=3, ***rank***(HC_1)= 4, and ***rank***(HC_3) = 5.

Note that if two or more components are tied at the same position in Eqs. (6) or (8), they have the same position. Figure 2(b) is an example of tired positions with the following relationships:

$$\underbrace{VS_{6A}^V}_{\text{Position 1}} \subset \underbrace{VS_{1A}^V = VS_{3A}^V = VS_{7A}^V}_{\text{Position 2 (tied)}} \subset \underbrace{VS_{2A}^V = VS_{5A}^V}_{\text{Position 3 (tied)}} \subset \underbrace{VS_{4A}^V}_{\text{Position 4}} \subset \underbrace{VS_{0A}^V}_{\text{Position 5}} \qquad (10)$$

$$\underbrace{HS_{0A}^H = HS_{2A}^H = HS_{6A}^H}_{\text{Position 1 (tied)}} \subset \underbrace{HS_{5A}^H}_{\text{Position 2}} \subset \underbrace{HS_{4A}^H}_{\text{Position 3}} \subset \underbrace{HS_{3A}^H}_{\text{Position 4}} \subset \underbrace{HS_{1A}^H}_{\text{Position 5}} \qquad (11)$$

The ranks of the vertical components in Figure 2(b) are as follows: ***rank***(VC_1) = 1, ***rank***(VC_3) = 2, ***rank***(VC_4) = 3, ***rank***(VC_5) = 4, ***rank***(VC_0) = ***rank***(VC_2) = ***rank***(VC_6) = 5. The ranks of the horizontal components are as follows: ***rank***(HC_0) = 1, ***rank***(HC_4) = 2, ***rank***(HC_2) = ***rank***(HC_5) = 3, ***rank***(HC_1) = ***rank***(HC_3) = ***rank***(HC_7) = 4, and ***rank***(HC_6) = 5.

Criterion 2: If a celled decision region is realizable by a TLP, the weights of the second layer of the TLP are determined by the following recursive formulas:

$$w_{vi} = \textbf{\textit{rank}}(HC_{i-1}) - \textbf{\textit{rank}}(HC_i) \text{ for vertical weights} \qquad (12)$$

$$w_{hj} = \textbf{\textit{rank}}(VC_{j-1}) - \textbf{\textit{rank}}(VC_j) \text{ for horizontal weights} \qquad (13)$$

where m is the number of the vertical weights and n is the number of the horizontal weights.

3 The Algorithm and Examples

3.1 Procedure of the algorithm

Step 1 (Feasibility-Determining Step): Determine the feasibility of a celled decision region by Criterion 1. If it is not realizable, stop the algorithm and conclude the decision region cannot be implemented by TLPs. If it is realizable, go to Step 2.

Step 2 (Weight-Selecting Step): Determine the weights of the second layer for a particular TLP by Criterion 2.

Step 3 (Threshold-Determination Step): θ_h is set to be equal to the minimum θ value of the celled sub-regions belonging to class A.

3.2 Examples of the Algorithm

Figure 1 is an example of an unrealizable celled decision region since one cannot find a particular order for the vertical strips to satisfy Eq. (6).

Figure 2(a) is an example of a realizable celled decision region without tied position where the ranks of the vertical and horizontal components are indicated early. By Criterion 2, the weights of the second layer of the TLP are determined as follows:

$w_{v1} = \textbf{\textit{rank}}(HC_0) - \textbf{\textit{rank}}(HC_1) = 1 - 4 = -3$,
$w_{v2} = \textbf{\textit{rank}}(HC_1) - \textbf{\textit{rank}}(HC_2) = 4 - 2 = 2$,
$w_{v3} = \textbf{\textit{rank}}(HC_2) - \textbf{\textit{rank}}(HC_3) = 2 - 5 = -3$,
$w_{v4} = \textbf{\textit{rank}}(HC_3) - \textbf{\textit{rank}}(HC_4) = 5 - 3 = 2$
$w_{h1} = \textbf{\textit{rank}}(VC_0) - \textbf{\textit{rank}}(VC_1) = 5 - 2 = 3$,
$w_{h2} = \textbf{\textit{rank}}(VC_1) - \textbf{\textit{rank}}(VC_2) = 2 - 1 = 1$,
$w_{h3} = \textbf{\textit{rank}}(VC_2) - \textbf{\textit{rank}}(VC_3) = 1 - 3 = -2$, and
$w_{h4} = \textbf{\textit{rank}}(VC_3) - \textbf{\textit{rank}}(VC_4) = 3 - 4 = -1$,

The decision region is implemented by letting $\theta_h = 0$.

Figure 2(b) is another realizable decision region with tied positions. The vertical and horizontal weights to implement the decision region are determined as follows: $w_{v1} = -3$, $w_{v2} = 1$, $w_{v3} = -1$, $w_{v4} = 2$, $w_{v5} = -1$, $w_{v6} = -2$, $w_{v7} = 1$, $w_{h1} = 4$, $w_{h2} = -4$, $w_{h3} = 3$, $w_{h4} = -1$, $w_{h5} = -1$, and $w_{h6} = -1$. By letting $\theta_h = 0$, one can implement the decision region by the above weights.

4 Conclusions

The author presented a constructive algorithm to examine the feasibility of implementation of celled decision regions by TLPs. If the celled decision region is realizable by a particular TLP, the algorithm generates the weights of the second layer of the TLP to implement the celled decision region without any training procedure.

The algorithm is fast based on two aspects. First, it tests the feasibility and determines the weights without any training procedure. Second, the classifications of input patterns are based on integer manipulations since the weights determined by the algorithm are all integers. The proposed algorithm consists of only three simple steps and is implemented easily by computer programming languages.

The partitioning capabilities of two-layer perceptrons for more complex decision regions such as convex or even non-convex decision regions might be interesting issues for the future work.

References

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

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

[3] P. J. Zwietering, E. H. L. Arts, and J. Wessels, Exact Classification with Two-Layered Perceptrons, *Int. Journal of Neural Systems*, Vol. 3, No. 2, 1992, pp. 143-156.

[4] G. Cybenko, Approximation by Superpositions of a Sigmoidal Function, *Math. Contr., Signals, Syst.*, 1989, pp. 303-314.

[5] 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.

[6] C. Lin and A. El-Jaroudi, A Study on the Partitioning Capabilities of Two-Layer Neural Networks, *IEEE Int. Conf. on Neural Network*, Orlando, Florida, July 1994, pp. I-360-365.

[7] G. J. Gibson and C. F. Cowan, On the Decision Regions of Multilayer Perceptrons, *Proceeding of the IEEE*, Vol. 78, No. 10, 1990, pp. 1590-1594.

[8] C. Lin and 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] G. J. Gibson, A Combinatorial Approach to Understanding Perceptron Capabilities, *IEEE Trans. on Neural Networks*, Vol. 4, No. 6, 1993, pp. 989-992.

[10] A Wieland and R. Leighton, Geometric Analysis of Neural Network Capabilities, *IEEE Int. Conf. on Neural Networks*, 1987, pp. III-385-392.

Proceedings of the 7th WSEAS International Conference on Neural Networks, Cavtat, Croatia, June 12-14, 2006 (pp126-131)

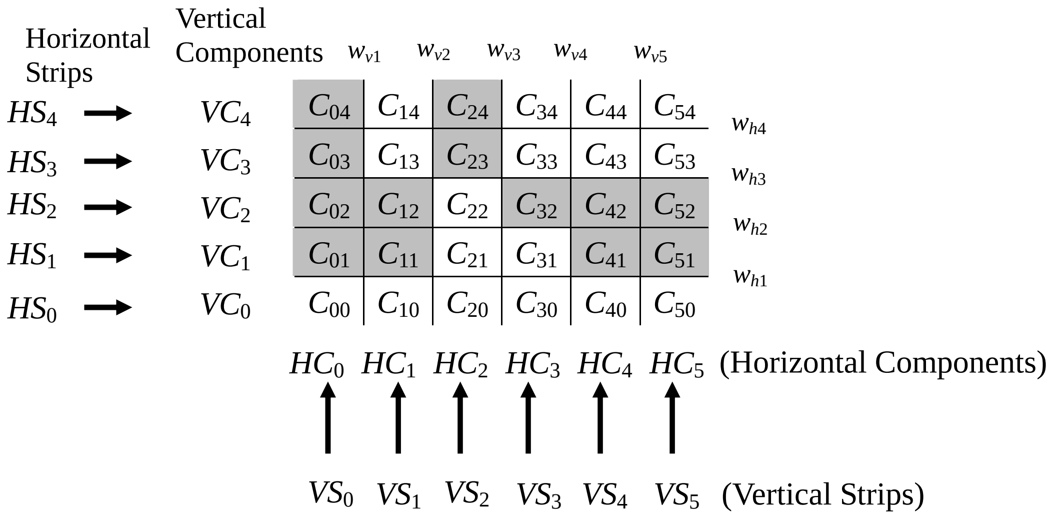


Figure 1: The Celled Decision Region

Remarks:

<u>For vertical strips</u>:

$VS_0 = \{C_{00}, C_{01}, C_{02}, C_{03}, C_{04}\}$, $VS_1 = \{C_{10}, C_{11}, C_{12}, C_{13}, C_{14}\}$,
$VS_2 = \{C_{20}, C_{21}, C_{22}, C_{23}, C_{24}\}$, $VS_3 = \{C_{30}, C_{31}, C_{32}, C_{33}, C_{34}\}$,
$VS_4 = \{C_{40}, C_{41}, C_{42}, C_{43}, C_{44}\}$, $VS_5 = \{C_{50}, C_{51}, C_{52}, C_{53}, C_{54}\}$.

The subsets of class A of vertical strips:

$VS_{0A}=\{C_{01}, C_{02}, C_{03}, C_{04}\}$, $VS_{1A} = \{C_{11}, C_{12}\}$, $VS_{2A} = \{C_{23}, C_{24}\}$,
$VS_{3A} = \{C_{32}\}$, $VS_{4A} = \{C_{41}, C_{42}\}$, $VS_{5A} = \{C_{51}, C_{52}\}$.

The sets of vertical components of VS_{iA}^{V}**:**

$VS_{0A}^{V} = \{VC_1, VC_2, VC_3, VC_4\}$, $VS_{1A}^{V} = \{VC_1, VC_2\}$, $VS_{2A}^{V} = \{VC_3, VC_4\}$,
$VS_{3A}^{V} = \{VC_2\}$, $VS_{4A}^{V} = \{VC_1, VC_2\}$, $VS_{5A}^{V} = \{VC_1, VC_2\}$.

<u>For horizontal strips:</u>

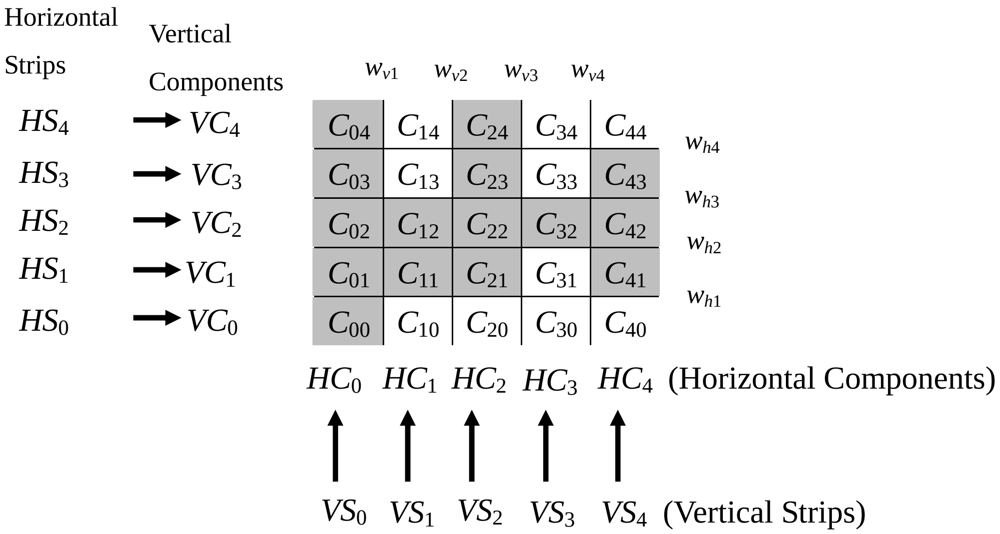
$HS_0 = \{C_{00}, C_{10}, C_{20}, C_{30}, C_{40}, C_{50},\}$, $HS_1 = \{C_{01}, C_{11}, C_{21}, C_{31}, C_{41}, C_{51},\}$,
$HS_2 = \{C_{02}, C_{12}, C_{22}, C_{32}, C_{42}, C_{52},\}$, $HS_3 = \{C_{03}, C_{13}, C_{23}, C_{33}, C_{43}, C_{53},\}$,
$HS_4 = \{C_{04}, C_{14}, C_{24}, C_{34}, C_{44}, C_{54}\}$.

The subsets of class A of horizontal strips:

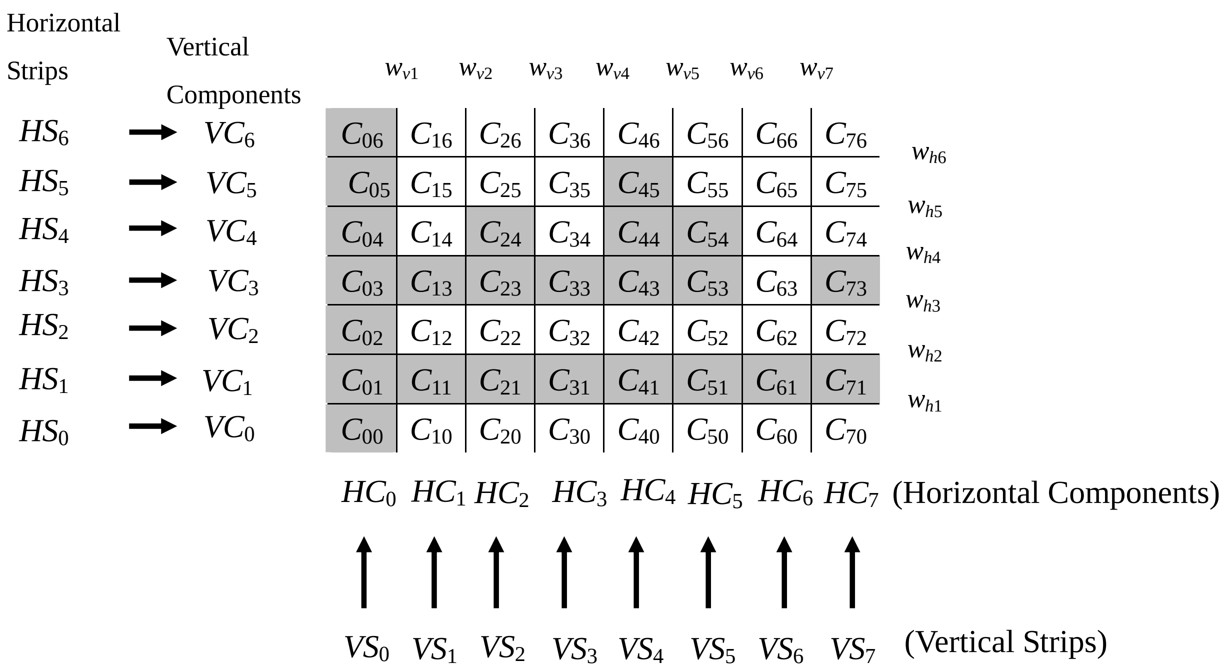
$HS_{0A} = \Phi$, $HS_{1A} = \{C_{01}, C_{11}, C_{41}, C_{51}\}$, $HS_{2A} = \{C_{02}, C_{12}, C_{32}, C_{42}, C_{52},\}$,
$HS_{3A} = \{C_{03}, C_{23}\}$, $HS_{4A} = \{C_{04}, C_{24}\}$.

The sets of horizontal components of HS_{jA}^{H}**:**

$HS_{0A}^{H} = \Phi$, $HS_{1A}^{H} = \{HC_0, HC_1, HC_4, HC_5\}$, $HS_{2A}^{H} = \{HC_0, HC_1, HC_3, HC_4, HC_5\}$,
$HS_{3A}^{H} = \{HC_0, HC_2\}$, $HS_{4A}^{H} = \{HC_0, HC_2\}$.



(a) Example without Ties of Positions of Vertical Strips



(b) Example with Ties of Positions of Vertical strips

Figure 2: The Examples of the Algorithm