

Jaringan Asosiasi

Randy Cahya Wihandika, S.ST., M.Kom.

Topik Hari Ini

- *Heteroassociative memory*
- Hebb rule

Asosiasi

- Pada proses pelatihan, terjadi pembentukan **asosiasi** antara hal-hal yang berbeda
- Contoh: bau roti di toko roti, warna, wajah



Sutan Sjahrir (1909–1966)



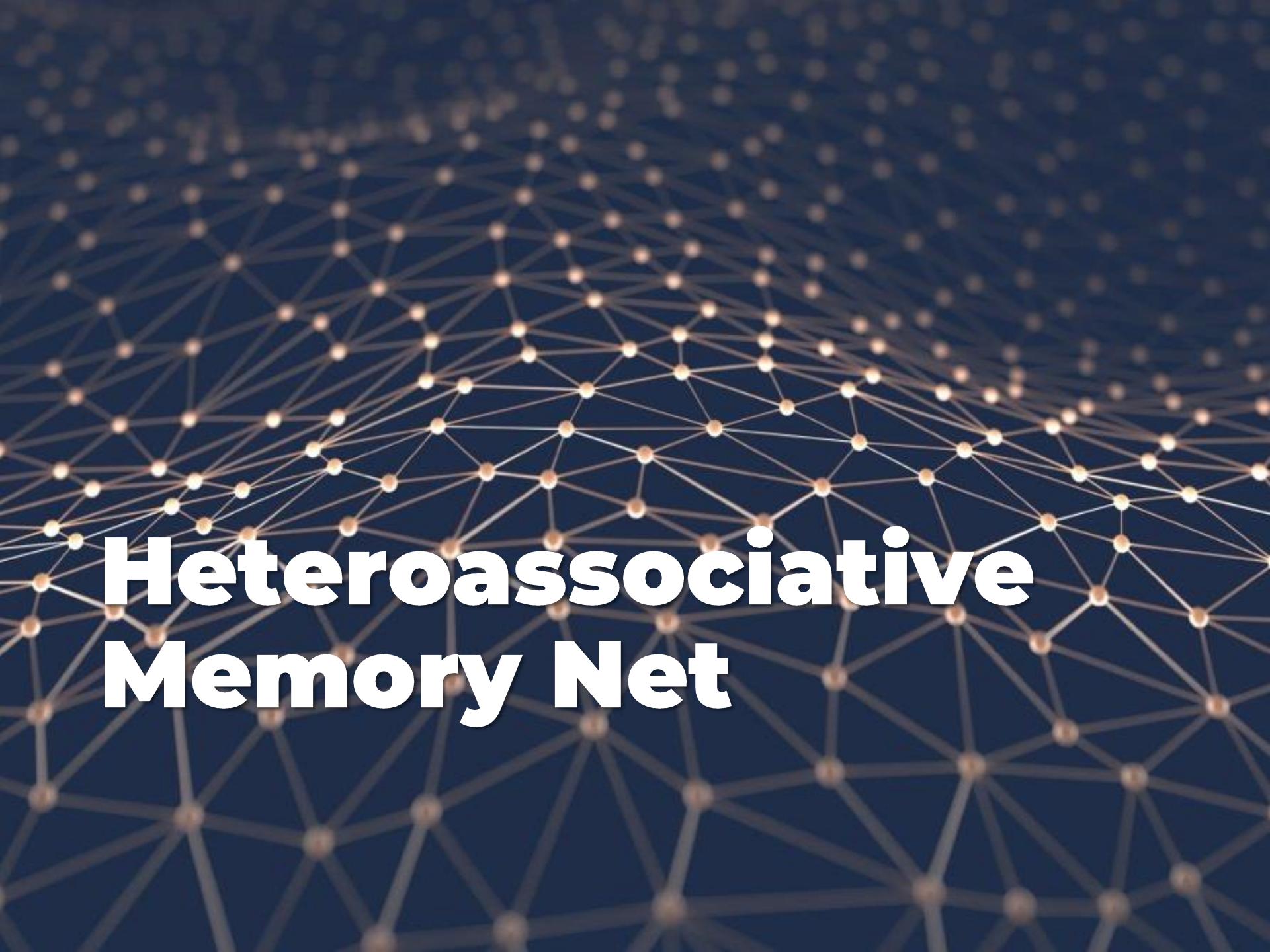
Yang manakah Sutan Sjahrir?

Asosiasi

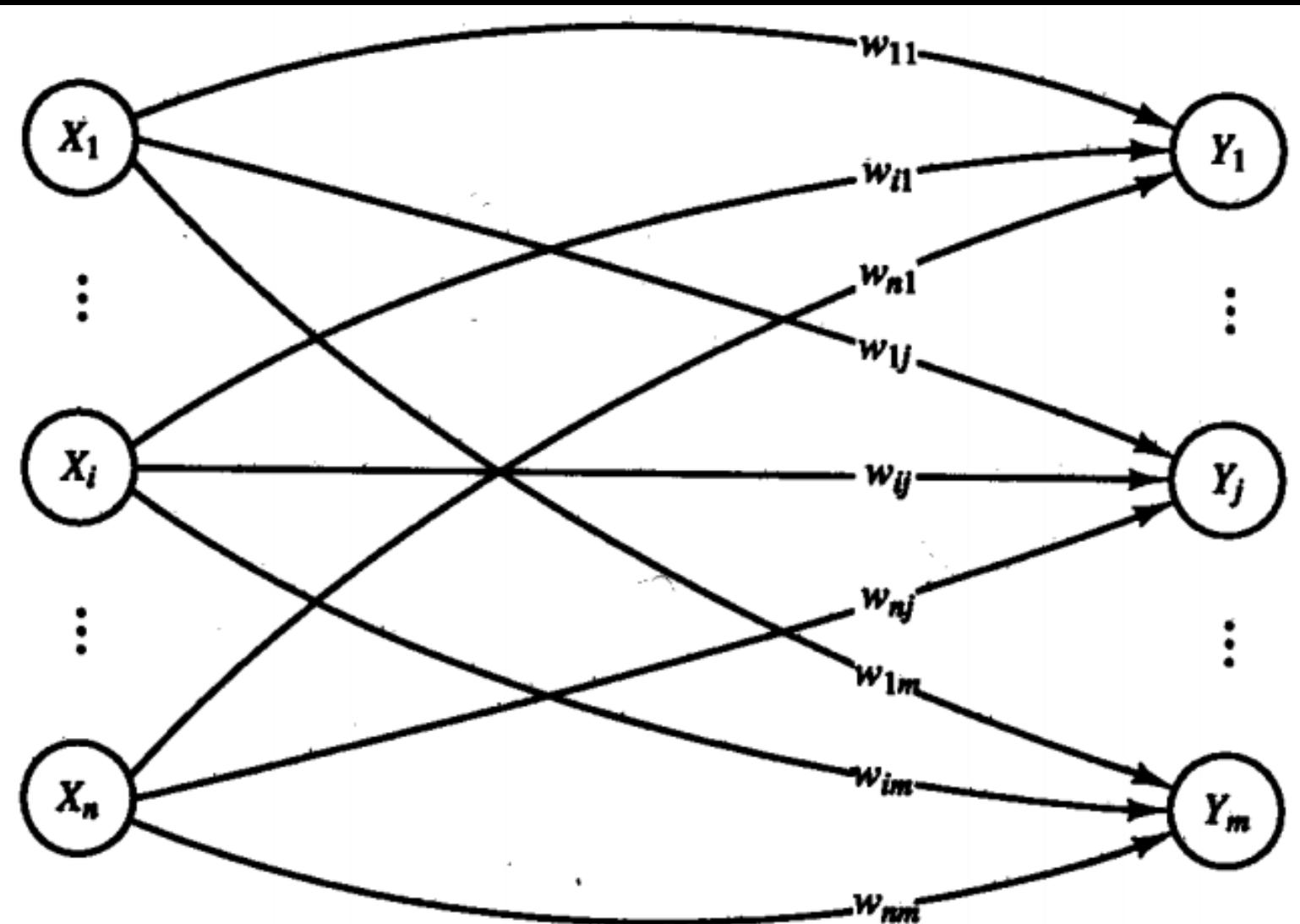
- Sebuah jaringan *associative memory* memodelkan memori manusia
- Asosiasi terjadi antara pasangan vektor s dan t
- Jika s sama dengan t , maka jaringan yang terbentuk dinamakan *autoassociative memory*
- Jika tidak, maka dinamakan *heteroassociative memory*

Asosiasi

- Algoritme pelatihan yang dapat digunakan:
Hebb rule dan delta rule



Heteroassociative Memory Net



**Input
Units**

**Output
Units**

Sumber: Fausett (1994)

Algoritme Pelatihan Hebb Rule

1. Inisialisasi semua bobot dengan nilai 0
2. Untuk setiap data latih s dan target t , lakukan langkah 3–5

3. Set nilai aktivasi setiap neuron input:

$$x_i = s_i$$

4. Set nilai aktivasi neuron *output*:

$$y_j = t_j$$

5. Ubah nilai bobot:

$$w'_{ij} = w_{ij} + x_i y_j$$

Algoritme Pelatihan Delta Rule

(Buka materi Adaline)

4. Hitung nilai y_{in} :

$$y_{in} = b + \sum_i x_i w_i$$

5. Update nilai bobot:

$$b' = b + \alpha(t - y_{in})$$

$$w'_i = w_i + \alpha(t - y_{in})x_i$$

$$\Delta w = \alpha(t - y_{in})x_i$$

Algoritme Heteroassociative Memory Net

1. Inisialisasi bobot
2. Untuk setiap vektor input, lakukan langkah 3–5
3. Set nilai vektor input ke neuron input
4. Hitung y_{in} :

$$y_{in\ j} = \sum_i x_i w_{ij}$$

Algoritme Heteroassociative Memory Net

5. Hitung nilai aktivasi (untuk target bipolar):

$$y_j = \begin{cases} 1 & \text{jika } y_{in_j} > 0 \\ 0 & \text{jika } y_{in_j} = 0 \\ -1 & \text{jika } y_{in_j} < 0 \end{cases}$$

Untuk target biner:

$$y_j = \begin{cases} 1 & \text{jika } x > 0 \\ 0 & \text{jika } x \leq 0 \end{cases}$$

Contoh

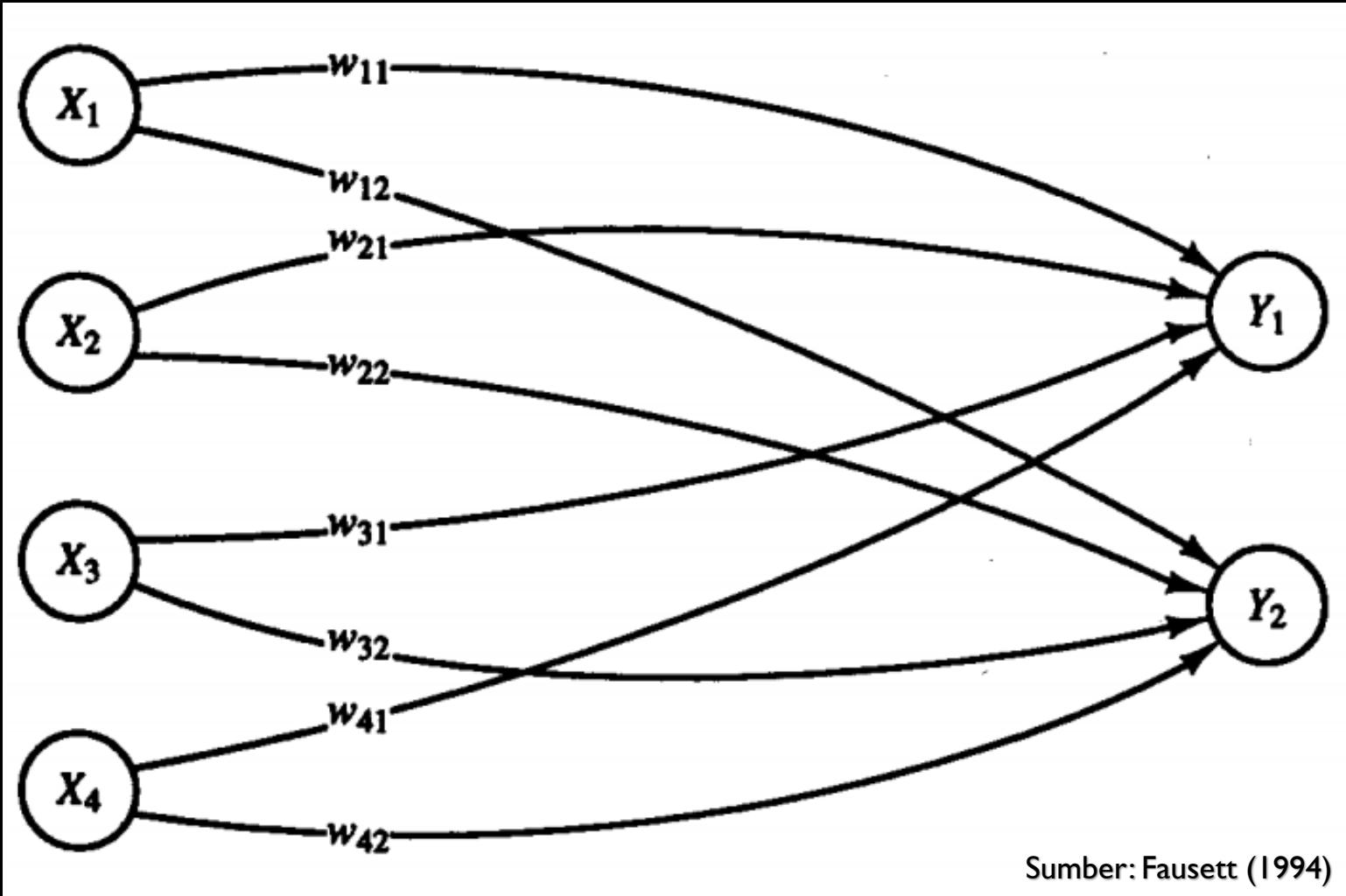
s

1. $(1, 0, 0, 0)$
2. $(1, 1, 0, 0)$
3. $(0, 0, 0, 1)$
4. $(0, 0, 1, 1)$

t

1. $(1, 0)$
2. $(1, 0)$
3. $(0, 1)$
4. $(0, 1)$

Contoh



Sumber: Fausett (1994)

Heteroassociative Memory Net

```
import numpy as np

def binstep(y, th=0):
    return [1 if i > th else 0 for i in y]

def hebb_assoc_train(s, t):
    w = np.zeros((len(s[0]), len(t[0])))
    for r, row in enumerate(s):
        for c, col in enumerate(row):
            w[c] = [w[c, i] + col * t[r, i] for i in range(len(t[r]))]
    return w
```

Heteroassociative Memory Net

```
def hebb_assoc_test(x, w):
    y = [np.dot(x, w[:, i]) for i in range(len(w[0]))]
    return binstep(y)

s = [[1, 0, 0, 0],
      [1, 1, 0, 0],
      [0, 0, 0, 1],
      [0, 0, 1, 1]]

t = np.array([[1, 0],
              [1, 0],
              [0, 1],
              [0, 1]])

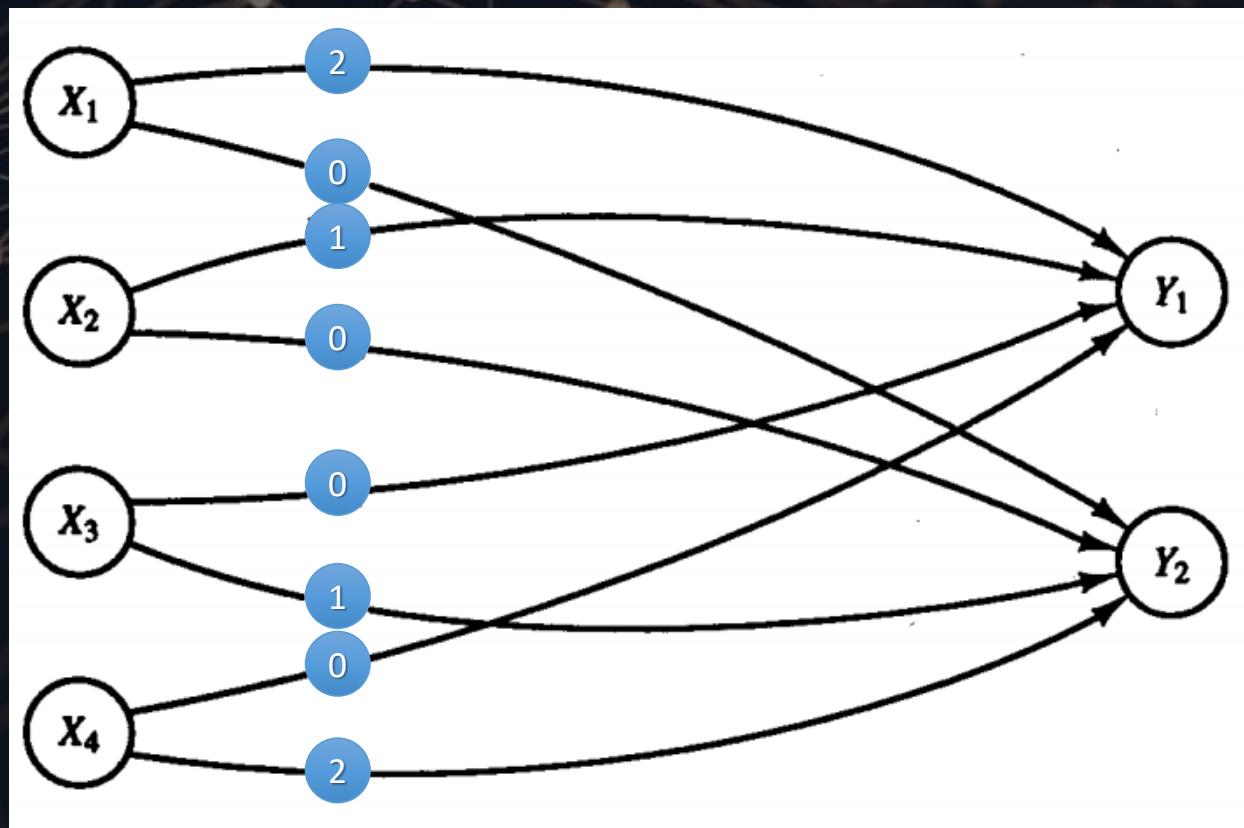
w = hebb_assoc_train(s, t)
y = hebb_assoc_test([0, 0, 1, 1], w)

print(w)
print(y)
```

Heteroassociative Memory Net

- Bobot pelatihan yang didapatkan:

2 0
1 0
0 1
0 2



Heteroassociative Memory Net

- Vektor pelatihan yang digunakan:
 $(1, 0, 0, 0) \rightarrow (1, 0)$
 $(1, 1, 0, 0) \rightarrow (1, 0)$
 $(0, 0, 0, 1) \rightarrow (0, 1)$
 $(0, 0, 1, 1) \rightarrow (0, 1)$
- Pengujian dengan vektor yang **mirip**:
 $(0, 1, 0, 0)$ memberikan hasil $(1, 0)$
- Pengujian dengan vektor yang **tidak mirip**:
 $(0, 1, 1, 0)$ memberikan hasil $(1, 1)$ (pola tidak dikenali)

Pelatihan Menggunakan Outer Product

- Nilai bobot dapat dihitung menggunakan *outer product*

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \begin{bmatrix} v_1 & v_2 \end{bmatrix} = \begin{bmatrix} u_1 v_1 & u_1 v_2 \\ u_2 v_1 & u_2 v_2 \\ u_3 v_1 & u_3 v_2 \end{bmatrix}$$

Pelatihan Menggunakan Outer Product

- $s_1 = (1, 0, 0, 0)$
- $t_1 = (1, 0)$

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} [1 \ 0] = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Pelatihan Menggunakan Outer Product

- $s_2 = (1, 1, 0, 0)$
- $t_2 = (1, 0)$

$$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} [1 \quad 0] = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Pelatihan Menggunakan Outer Product

- $s_3 = (0, 0, 0, 1)$
- $t_3 = (0, 1)$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} [0 \quad 1] = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

Pelatihan Menggunakan Outer Product

- $s_4 = (0, 0, 1, 1)$
- $t_4 = (0, 1)$

$$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} [0 \quad 1] = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Pelatihan Menggunakan Outer Product

$$W = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 2 \end{bmatrix}$$

Pelatihan Menggunakan Outer Product

```
def hebb_assoc_train_outer(s, t):
    p = [np.outer(np.reshape(s[i], (-1, 1)), t[i]) for
i in range(len(s))]
    return np.sum(p, 0)
```

Pengujian Menggunakan Perkalian Matriks

- Pengujian juga dapat dilakukan dengan perkalian matriks

Pengujian Menggunakan Perkalian Matriks

- $s_3 = (1,0,0,0)$

$$s_3 w = (1,0,0,0) \begin{bmatrix} 2 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 2 \end{bmatrix} = (2,0)$$
$$y = (1,0)$$

Pengujian Menggunakan Perkalian Matriks

```
def hebb_assoc_test_mat(x, w):  
    return binstep(np.matmul(x, w))
```



Alhamdulillah.