

Kecerdasan Komputasional

Dr. Indah Agustien S.,S.Kom., M.Kom.
Universitas Trunojoyo Madura

Jaringan Syaraf Tiruan



1

Jaringan Syaraf Tiruan - Content

1 Jaringan Syaraf Tiruan

2 Neuron

3 Perceptron

4 Supervised Learning

5 Unsupervised Learning

2

Unsupervised Learning

1

Data Training : Sinyal input / feature

2

Tidak terdapat data target yang berfungsi sebagai supervisor

3

Klasifikasi atau pengelompokkan berdasarkan kesamaan fitur

Kecerdasan Komputasional | Jaringan Syaraf Tiruan

3

K-Self Organizing Map

Memetakan data input yang berdimensi besar menjadi berdimensi dua

Membentuk 'Peta' : rectangular atau hexagonal grid dua dimensi

$$\begin{bmatrix} x_{11} & \dots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \dots & x_{mn} \end{bmatrix}$$

Dataset berdimensi n , berjumlah m



Peta Data : 3 kelompok

Kecerdasan Komputasional | Jaringan Syaraf Tiruan

4

K-Self Organizing Map

1

Algoritma Pembelajaran : Competitive Learning

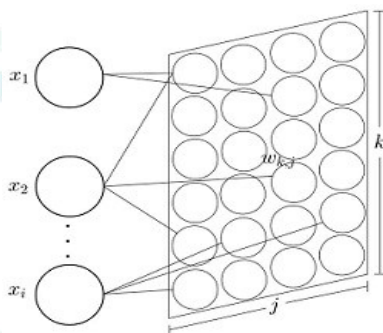
2

Bobot yang menuju atau dekat dengan winning neuron yang akan diupdate

Kecerdasan Komputasional | Jaringan Syaraf Tiruan

5

Arsitektur K-SOM

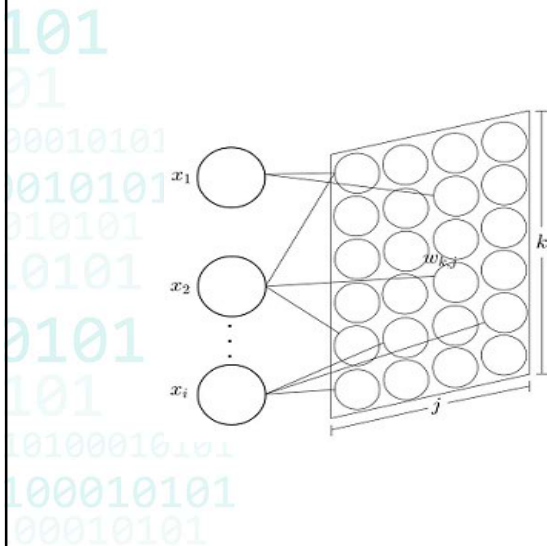


- Output K-SOM : Peta Grid 2D
- Tiap Grid adalah neuron output
- Neuron merepresentasikan kelompok output
- Jumlah kelas maksimal adalah sebanyak jumlah data

Kecerdasan Komputasional | Jaringan Syaraf Tiruan

6

Arsitektur K-SOM



x_1, \dots, x_i : sinyal input yang berdimensi i

w_{kj} : code vector, atau bobot, merupakan centroid dari suatu kelas atau kelompok

Kecerdasan Komputasional | Jaringan Syaraf Tiruan

7

Arsitektur K-SOM

Pengelompokan data berdasarkan kesamaan input

Winning Neuron pada output jika memiliki jarak paling minimal dengan sinyal input

Update dilakukan pada code vector yang dekat dengan winning neuron

Kecerdasan Komputasional | Jaringan Syaraf Tiruan

8

Algoritma Pembelajaran

- Inisialisasi bobot atau code vector
- Cari winning neuron, dengan mencari jarak minimal antara sinyal input dengan semua neuron pada output

$$w_{mn} = \min_{\forall k_j} (||w_{kj} - x_p||)$$

- Hitung nilai neighborhood, untuk menentukan code vector mana saja yang akan diupdate. Semakin jauh code vector terhadap winning neuron maka kemungkinan untuk update semakin kecil, begitu juga sebaliknya

Kecerdasan Komputasional | Jaringan Syaraf Tiruan

9

Algoritma Pembelajaran

Contoh fungsi Neighborhood:

$$h_{mn,kj}(t) = e^{-\frac{||c_{mn}-ckj||}{2\sigma^2(t)}}$$

- Update Code Vector :

$$w_{kj}(t+1) = w_{kj}(t) + \eta(t)h_{mn,kj}(t)[x_p - w_{kj}(t)]$$
- Lakukan langkah 2-4 sampai epoch tertentu atau Error tertentu :

$$\varepsilon_T = \sum_{p=1}^P ||x_p - w_{mn}(t)||$$

Kecerdasan Komputasional | Jaringan Syaraf Tiruan

10

K-Self Organizing Map

- Empat buah vector berdimensi 4, yaitu :
 - [1 1 1 0]
 - [0 0 0 1]
 - [1 0 0 0]
 - [0 0 1 1]
- Jumlah kelas/kelompok : 2
- Learning Rate = 0.5
- Fungsi Neighborhood adalah fungsi Gaussian, dengan Standar Deviasi = 2
- Code Vector kelas 1 =[0.9 0.7 0.1 0.2]
- Code Vector kelas 2 =[0.2 0.2 0.7 0.9]

Kecerdasan Komputasional | Jaringan Syaraf Tiruan

11

K-Self Organizing Map

Epoch – 1, data ke-1

- Cari Winning Neuron :
 - jarak antara [1 1 1 0] dengan [0.9 0.7 0.1 0.2]
 - Jarak antara [1 1 1 0] dengan [0.2 0.2 0.7 0.9]
 - Winning Neuron :
- Hitung Nilai Neighborhood
- Update Bobot/code vector

Kecerdasan Komputasional | Jaringan Syaraf Tiruan

12

K-Self Organizing Map

Epoch – 1, data ke-2

- Cari Winning Neuron :
 - jarak antara [0 0 0 1] dengan [? ? ? ?]
 - Jarak antara [0 0 0 1] dengan [? ? ? ?]
 - Winning Neuron :
- Hitung Nilai Neighborhood
- Update Bobot/code vector


END