



KOM 422

Pengantar Pemrosesan Bahasa Alami

Contoh Kasus: *An Automatic
Speaker Recognition System with
SOM Algorithm*

Ganjil 2019/2020

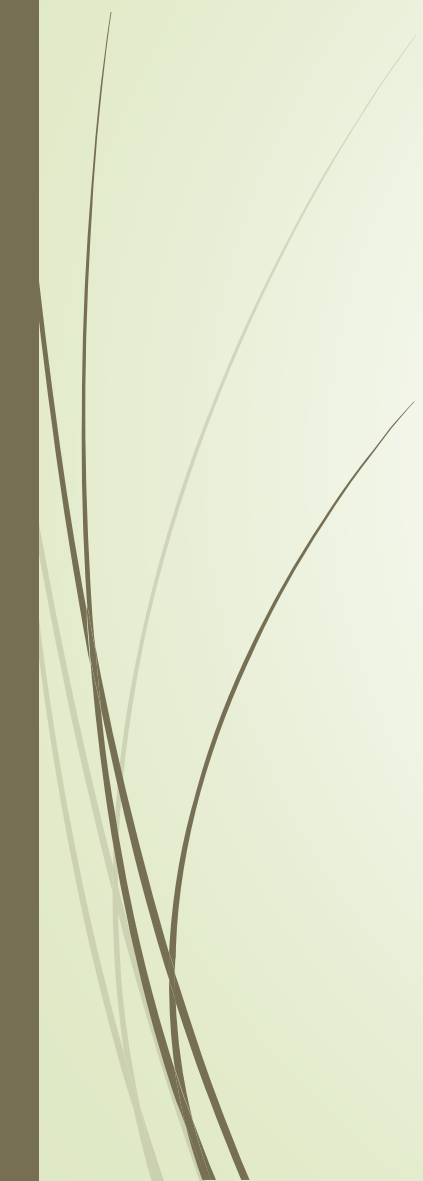


Overview

- Metode speaker recognition adalah penggunaan komputer untuk dimanfaatkan di kehidupan manusia terutama membantu dalam aspek pengindraan manusia melalui media suara
- Metode pada saat ini sudah dapat:
 - Mengenal kata yang diucapkan manusia
 - Mengenal pembicara
 - Mengenal bahasa yang diucapkan
- Khusus untuk mengenali pembicara(biometrik), suara dapat digunakan selain sidik jari dan retina mata karena suara juga memiliki karakteristik yang unik

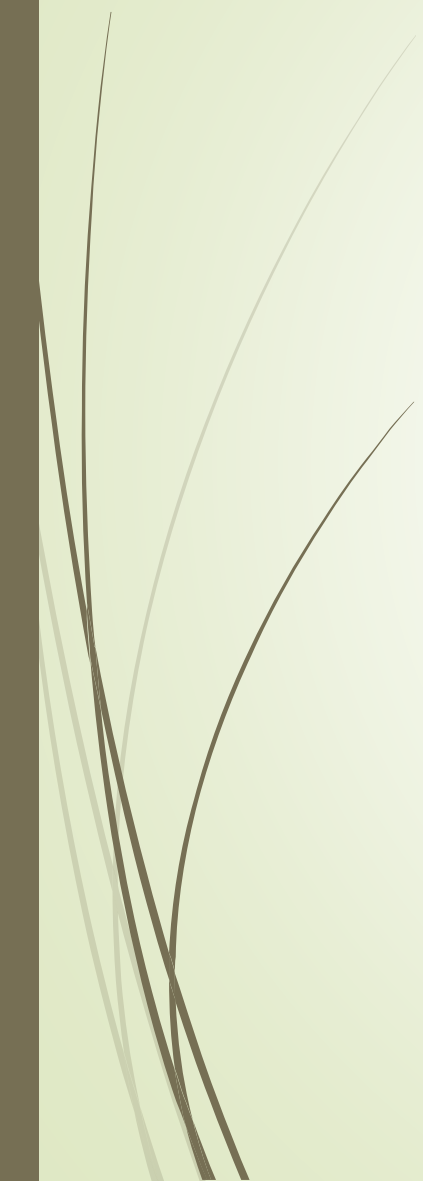


Speaker Recognition

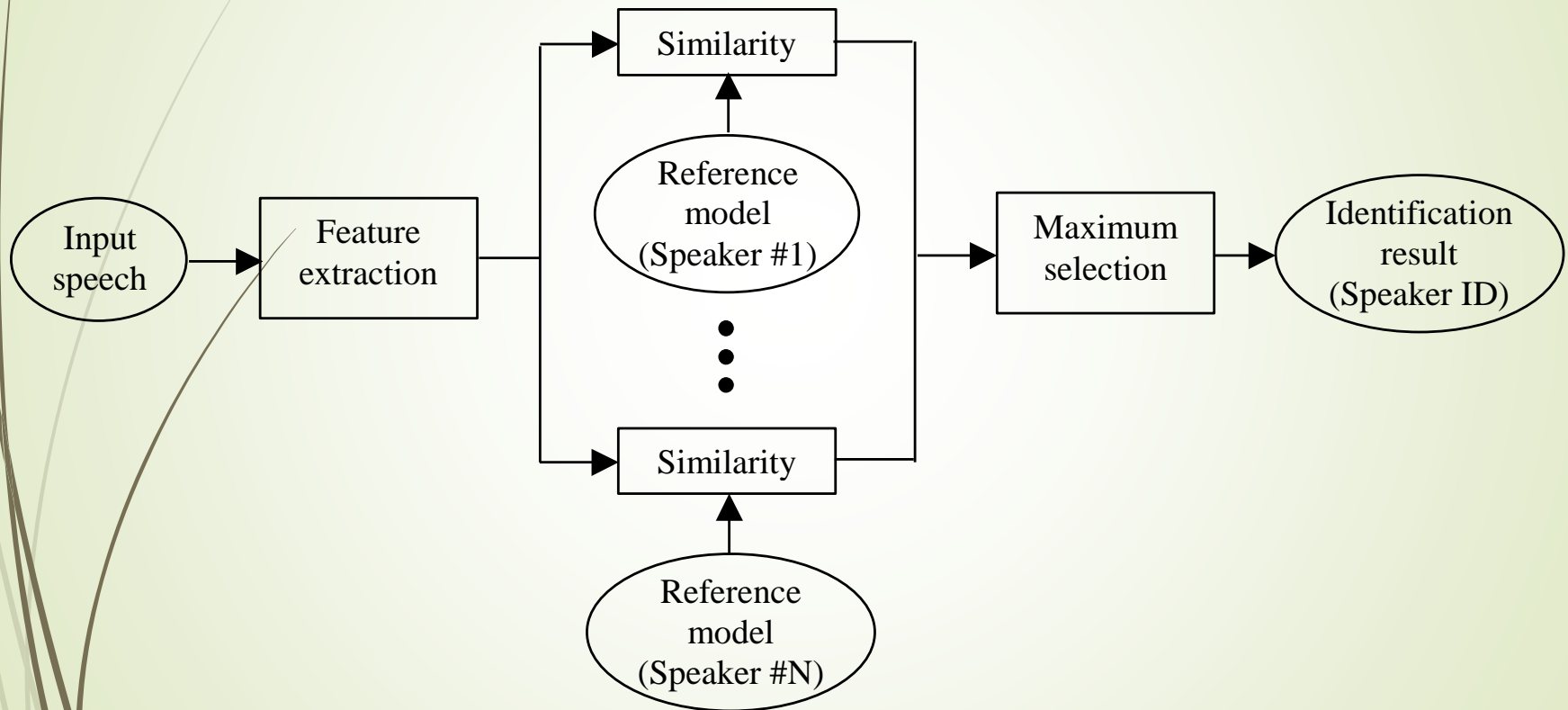
- *Speaker recognition* adalah proses mengenali secara otomatis siapa yang berbicara/berucap dengan menggunakan informasi individu yaitu jenis suara pembicara
 - Teknik ini memungkinkan proses verifikasi suara pembicara sesuai identitas, dan memiliki akses kendali (voice dialing, banking by telephone, telephone shopping, database access services, information services, voice mail, security control for confidential information areas, and remote access to computers)
- 



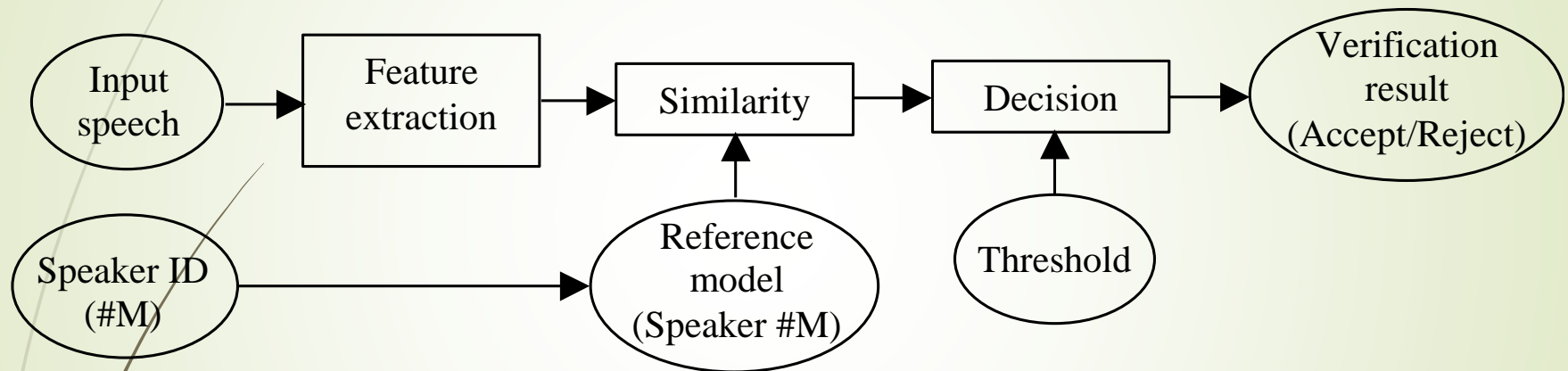
Principle of Speaker Recognition

- *Speaker recognition* dapat diklasifikasikan ke dalam identifikasi dan verifikasi
 - *Speaker identification* adalah proses menentukan pembicara yang mana yang sesuai dengan melihat pola suara yang diucapkannya
 - *Speaker verification*, adalah proses menerima atau menolak identitas pembicara
- 

Speaker Identification



Speaker Verification





Speaker Recognition

- Semua *speaker recognition systems* memiliki 2 fase yang secara umum sama:

1. Training phase (Enrolment)

Setiap speaker teregistrasi harus menyediakan sample dari suara mereka sehingga sistem dapat membangun/melatih model referensi untuk speaker tersebut

2. Testing phase (Operational)

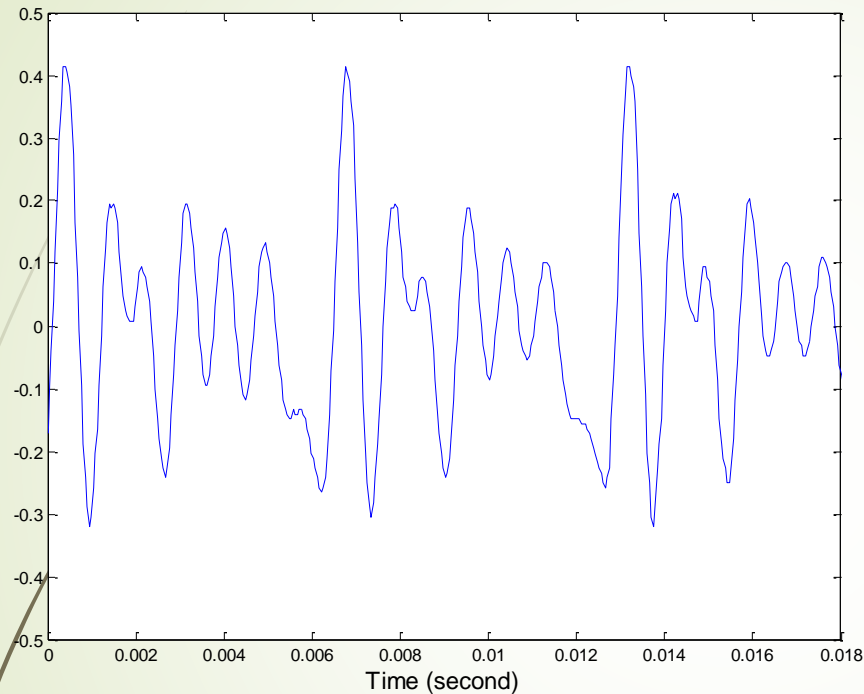
Input yang sesuai dengan model pembelajaran/referensi akan dikenali



Speech Feature Extraction

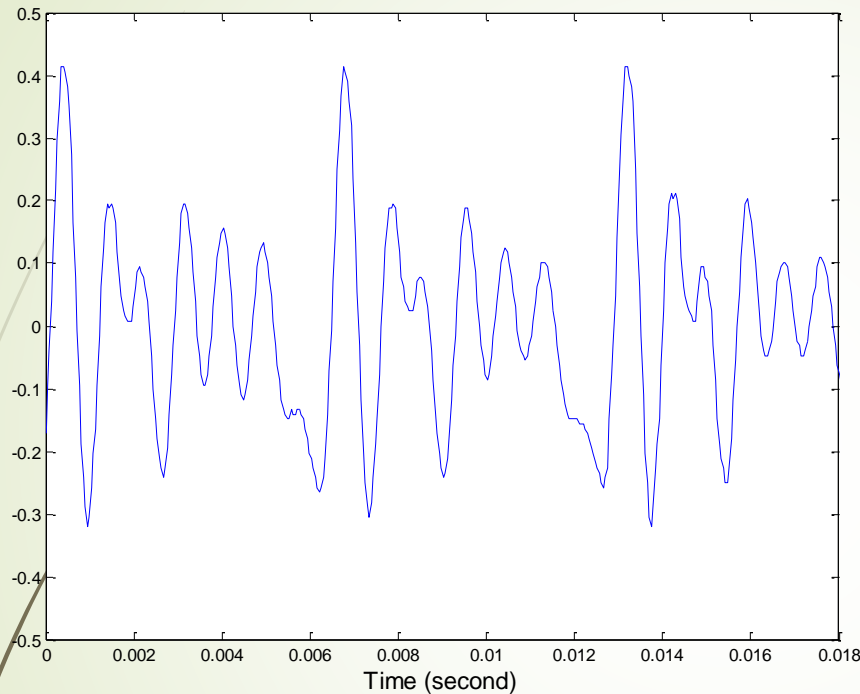
- Apa kegunaan dari ekstraksi fitur (ciri)?
- Konversi Gelombang Suara (Speech waveform) ke suatu himpunan fitur untuk analisis selanjutnya
- Proses ekstraksi fitur dianggap paling penting dalam sistem pengenalan suara yang mempunyai tujuan menangkap fitur untuk membedakan suara yang satu dengan suara yang lain
- Metode umum yang digunakan?
 - LPC (Linear Predictive Coding)
 - MFCC (Mel Frequency Cepstral Coefficients)
 - ZCPA (Zero Crossing Peak Amplitudo)

Speech Feature Extraction



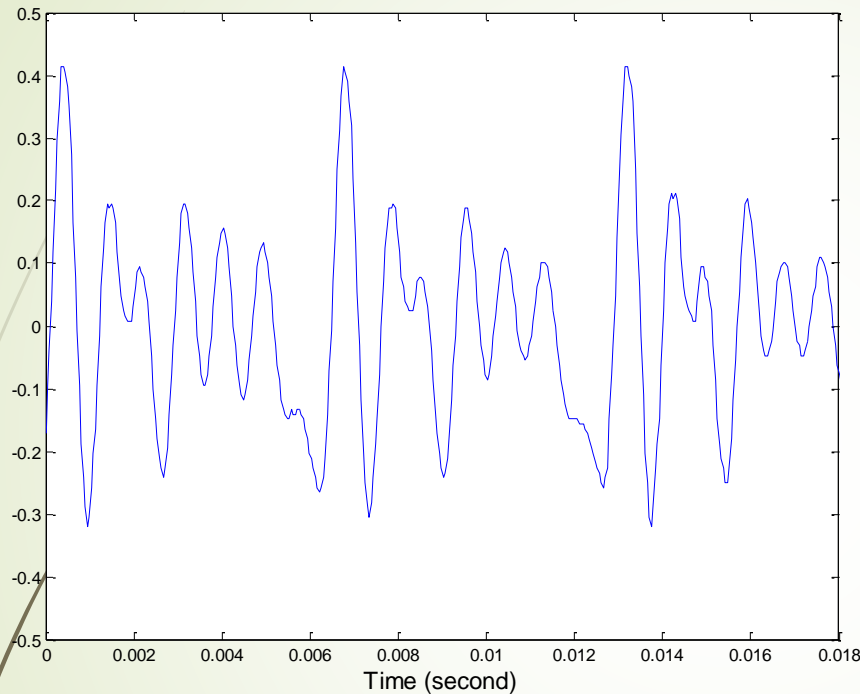
- Sinyal suara analog disebut sebagai quasi-stationary
- Apa Representasi dari sinyal suara?

Speech Feature Extraction



Gelombang suara yang ditangkap berupa sinyal analog. Sinyal analog harus diubah menjadi sinyal digital melalui proses digitalisasi. Proses digitalisasi suara terdiri atas dua tahap, yaitu *sampling* dan kuantisasi (Jurafsky & Martin 2000)

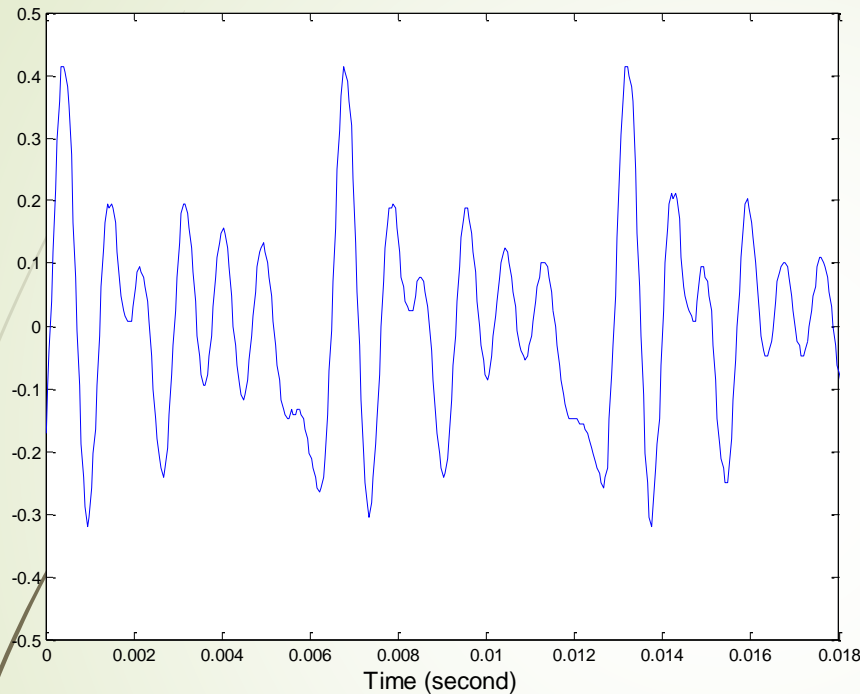
Speech Feature Extraction



Sampling adalah pengambilan nilai-nilai dalam jangka waktu tertentu.

Sampling rate yang biasanya digunakan pada pengenalan suara ialah 8000 Hz sampai dengan 16000 Hz (Jurafsky & Martin 2000).

Speech Feature Extraction



Hubungan panjang vektor yang dihasilkan, *sampling rate*, dan panjang data suara yang didigitalisasi dinyatakan dengan persamaan:

$$S = F_s \times T$$

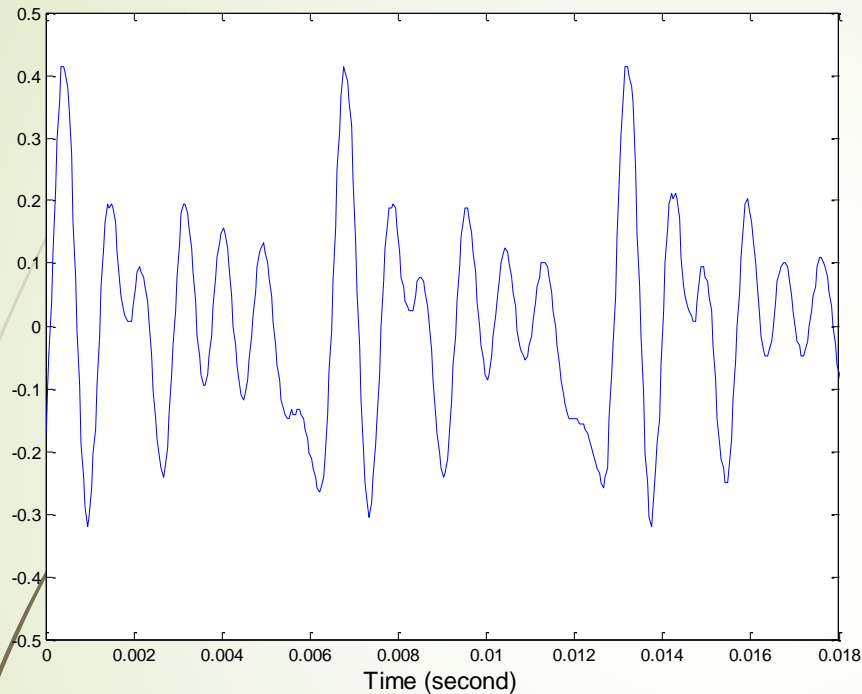
Keterangan:

S = panjang vektor

F_s = *sampling rate* yang digunakan (Hertz)

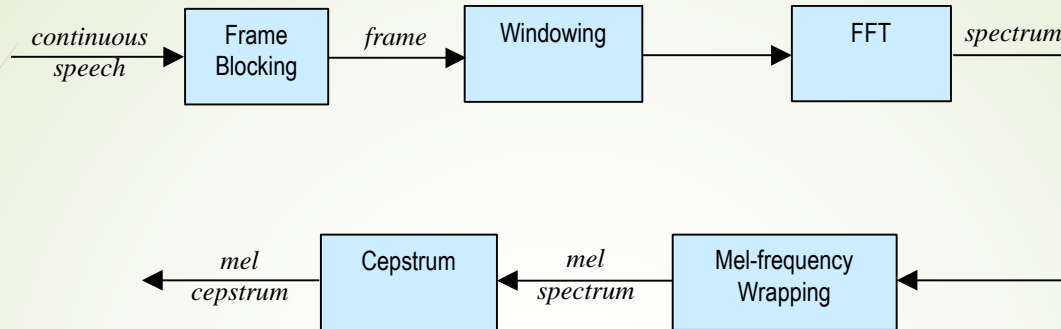
T = panjang suara (detik)

Speech Feature Extraction



Setelah tahap *sampling*, proses selanjutnya adalah proses kuantisasi. Proses ini menyimpan nilai amplitudo ke dalam representasi nilai 8 bit atau 16 bit (Jurafksy & Martin 2000).

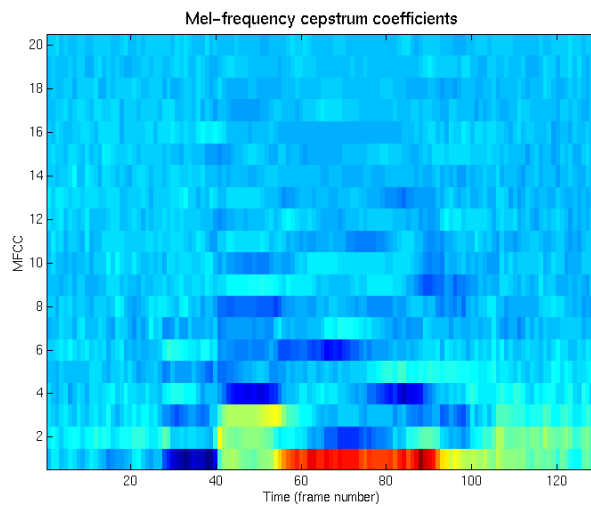
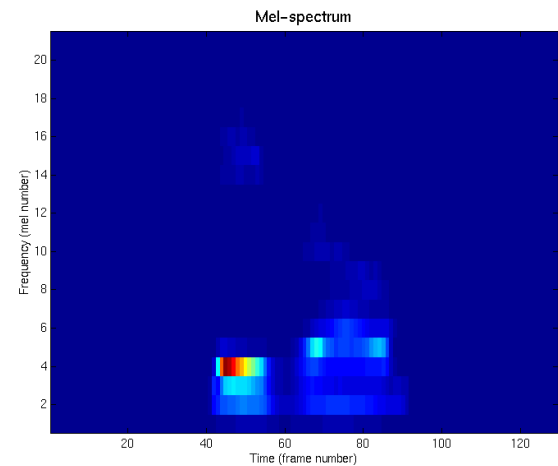
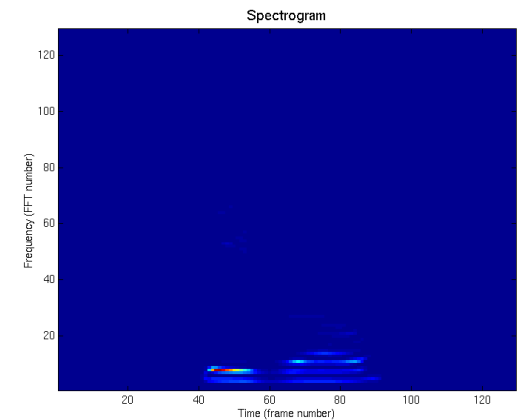
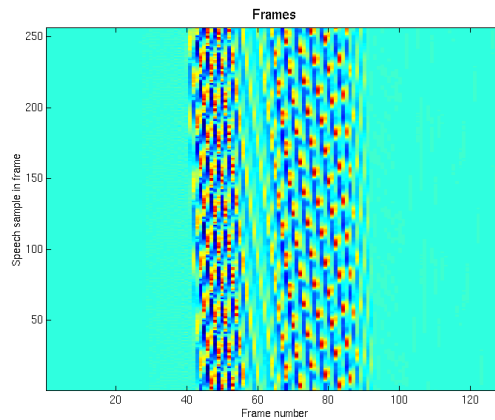
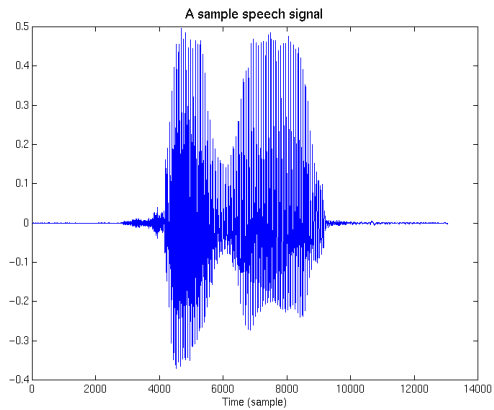
MFCC



Ekstraksi ciri merupakan suatu teknik yang digunakan untuk menghasilkan suatu vektor yang digunakan sebagai penciri dengan menghitung koefisien *cepstral*. Tahapan proses MFCC ialah sebagai berikut (Do 1994) :

- *Frame Blocking.*
- *Windowing.*
- *Fast Fourier Transform (FFT).*
- *Mel Frequency Wrapping.*
- *Cepstrum Coefficient.*

MFCC



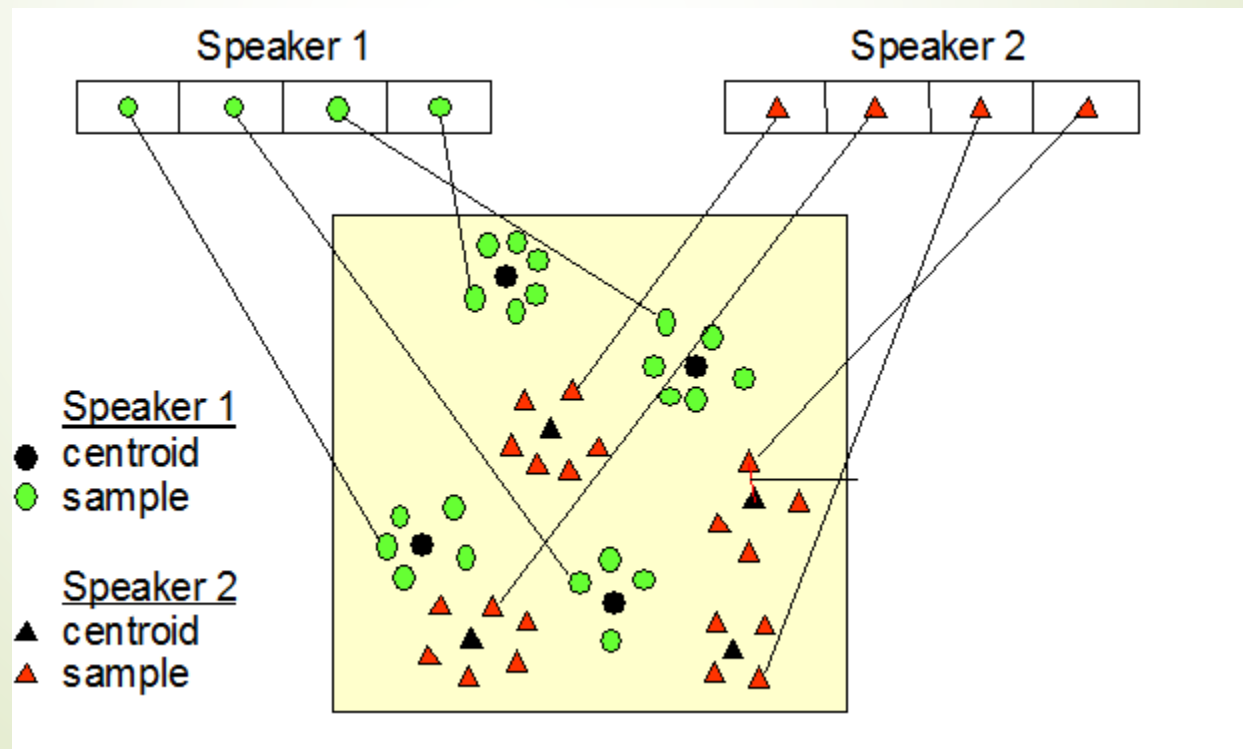


Pattern Recognition

- Speaker recognition → Pattern Recognition
- Tujuan dari pattern recognition adalah proses pengenalan melalui ciri/pola suatu obyek sehingga dapat dikategorikan ke dalam salah satu kelas
- Beberapa Algoritme untuk Feature Matching:
 - DTW, HMM, SVM, K-Means(Codebook), Self Organizing Map, Vector Quantization, dll

Pattern Recognition

- Proses Mapping vector.ciri ke suatu region terbatas. Setiap region disebut sebagai cluster dan direpresentasikan dengan pusat klaster.





Clustering

- Tujuan dari clustering adalah untuk menemukan model yang berbeda dan terbagi menjadi kelas sesuai dengan data yang ada dari masing-masing objek atau data yang kelasnya belum diketahui (unsupervised)
- Salah satu model klaster (unsupervised) dengan teknik Jaringan saraf tiruan adalah SOM (Jaringan Kohonen)
- Pada metode SOM akan membentuk suatu klaster, dimana neuron-neouran (input berupa ciri) akan membentuk kelasnya masing-masing secara otomatis berdasarkan nilai tertentu dari informasi input/ciri/fitur tersebut

SOM Algorithm

Training data: vectors, X

- Vectors of length n

$(x_{1,1}, x_{1,2}, \dots, x_{1,i}, \dots, x_{1,n})$

$(x_{2,1}, x_{2,2}, \dots, x_{2,i}, \dots, x_{2,n})$

...

$(x_{j,1}, x_{j,2}, \dots, x_{j,i}, \dots, x_{j,n})$

...

$(x_{p,1}, x_{p,2}, \dots, x_{p,i}, \dots, x_{p,n})$

p distinct training vectors

- Vector components are real numbers

Outputs

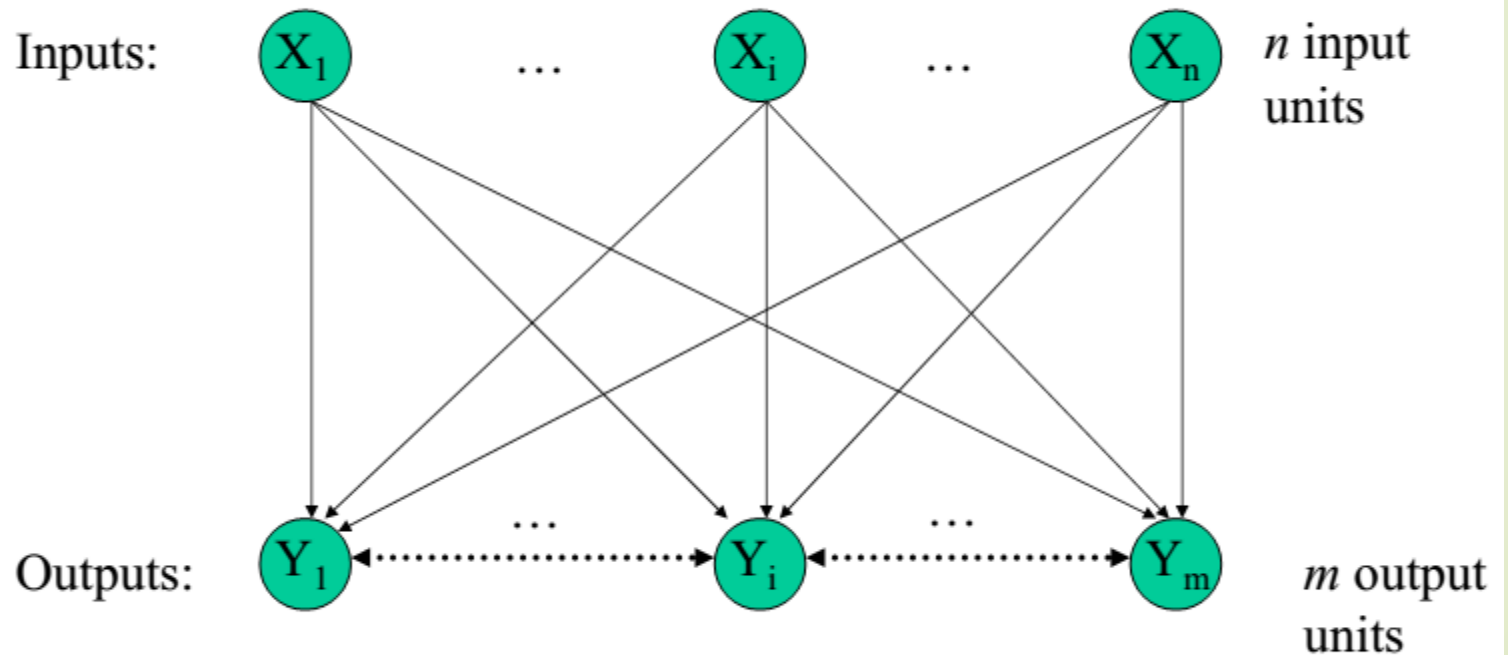
- A vector, Y , of length m : $(y_1, y_2, \dots, y_i, \dots, y_m)$
 - Sometimes $m < n$, sometimes $m > n$, sometimes $m = n$
- Each of the p vectors in the training data is classified as falling in one of m clusters or categories
- That is: Which category does the training vector fall into?

Generalization

- For a new vector: $(x_{j,1}, x_{j,2}, \dots, x_{j,i}, \dots, x_{j,n})$
- Which of the m categories (clusters) does it fall into?

SOM Algorithm

Network Architecture



Note: There is one weight vector of length n associated with each output unit

SOM Algorithm

- Select output layer network topology
 - Initialize current neighborhood distance, $D(0)$, to a positive value
- Initialize weights from inputs to outputs to small random values
- Let $t = 1$
- While computational bounds are not exceeded do

- 1) Select an input sample i_l
- 2) Compute the square of the Euclidean distance of i_l from weight vectors (w_j) associated with each output node

$$\sum_{k=1}^n (i_{l,k} - w_{j,k}(t))^2$$

- 3) Select output node j^* that has weight vector with minimum value from step 2)
- 4) Update weights to all nodes within a topological distance given by $D(t)$ from j^* , using the weight update rule:

$$w_j(t+1) = w_j(t) + \eta(t)(i_l - w_j(t))$$

- 5) Increment t
- Endwhile

Learning rate generally decreases with time:

$$0 < \eta(t) \leq \eta(t-1) \leq 1$$

From Mehotra et al. (1997), p. 189



SOM Algorithm

► Training

- Select output layer topology
- Train weights connecting inputs to outputs
- Topology is used, in conjunction with current mapping of inputs to outputs, to define which weights will be updated
- Distance measure using the topology is reduced over time; reduces the number of weights that get updated per iteration
- Learning rate is reduced over time

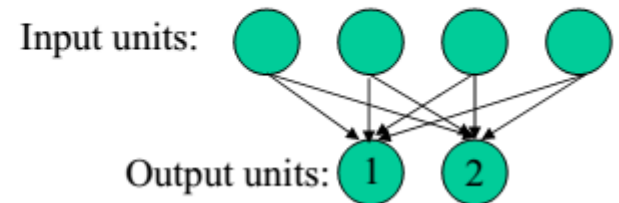
► Testing

- Use weights from training

Contoh

- From Fausett (1994)
- $n = 4, m = 2$
- Training samples
 - i1: (1, 1, 0, 0)
 - i2: (0, 0, 0, 1)
 - i3: (1, 0, 0, 0)
 - i4: (0, 0, 1, 1)

Network Architecture



Contoh

Euclidean Distances Between Data Samples

- Training samples

i1: (1, 1, 0, 0)

i2: (0, 0, 0, 1)

i3: (1, 0, 0, 0)

i4: (0, 0, 1, 1)

| | i1 | i2 | i3 | i4 |
|----|----|----|----|----|
| i1 | 0 | | | |
| i2 | 3 | 0 | | |
| i3 | 1 | 2 | 0 | |
| i4 | 4 | 1 | 3 | 0 |

Input units:



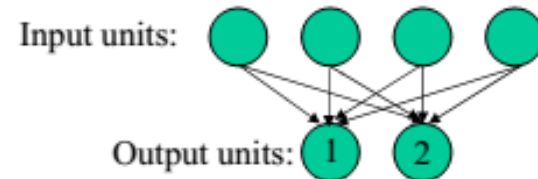
Output units:



What might we expect from the SOM?

Contoh

Example Details



- Training samples
 - i1: (1, 1, 0, 0)
 - i2: (0, 0, 0, 1)
 - i3: (1, 0, 0, 0)
 - i4: (0, 0, 1, 1)
- Let neighborhood = 0
 - Only update weights associated with winning output unit (cluster) at each iteration
- Learning rate
 - $\eta(t) = 0.6; 1 \leq t \leq 4$
 - $\eta(t) = 0.5 \eta(1); 5 \leq t \leq 8$
 - $\eta(t) = 0.5 \eta(5); 9 \leq t \leq 12$
 - etc.
- Initial weight matrix (random values between 0 and 1) $\left\{ \begin{array}{l} \text{Unit 1:} \\ \text{Unit 2:} \end{array} \right. \begin{bmatrix} .2 & .6 & .5 & .9 \\ .8 & .4 & .7 & .3 \end{bmatrix}$

$$d^2 = (\text{Euclidean distance})^2 = \sum_{k=1}^n (i_{l,k} - w_{j,k}(t))^2$$

$$\text{Weight update: } w_j(t+1) = w_j(t) + \eta(t)(i_l - w_j(t))$$

19

Problem: Calculate the weight updates for the first four steps

Contoh

First Weight Update

i1: (1, 1, 0, 0)

i2: (0, 0, 0, 1)

i3: (1, 0, 0, 0)

i4: (0, 0, 1, 1)

- Training sample: i1

$$\begin{array}{l} \text{Unit 1: } \begin{bmatrix} .2 & .6 & .5 & .9 \end{bmatrix} \\ \text{Unit 2: } \begin{bmatrix} .8 & .4 & .7 & .3 \end{bmatrix} \end{array}$$

- Unit 1 weights

- $d^2 = (.2-1)^2 + (.6-1)^2 + (.5-0)^2 + (.9-0)^2 = 1.86$

- Unit 2 weights

- $d^2 = (.8-1)^2 + (.4-1)^2 + (.7-0)^2 + (.3-0)^2 = .98$

- Unit 2 wins

- Weights on winning unit are updated

- Giving an updated weight matrix:

$$\begin{aligned} \text{new-unit2-weights} &= [.8 \ .4 \ .7 \ .3] + 0.6([1 \ 1 \ 0 \ 0] - [.8 \ .4 \ .7 \ .3]) = \\ &= [.92 \ .76 \ .28 \ .12] \end{aligned}$$

$$\begin{array}{l} \text{Unit 1: } \begin{bmatrix} .2 & .6 & .5 & .9 \end{bmatrix} \\ \text{Unit 2: } \begin{bmatrix} .92 & .76 & .28 & .12 \end{bmatrix} \end{array}$$

Contoh

Second Weight Update

i1: (1, 1, 0, 0)

i2: (0, 0, 0, 1)

i3: (1, 0, 0, 0)

i4: (0, 0, 1, 1)

- Training sample: i2

$$\begin{array}{l} \text{Unit 1: } \begin{bmatrix} .2 & .6 & .5 & .9 \end{bmatrix} \\ \text{Unit 2: } \begin{bmatrix} .92 & .76 & .28 & .12 \end{bmatrix} \end{array}$$

- Unit 1 weights

- $d^2 = (.2-0)^2 + (.6-0)^2 + (.5-0)^2 + (.9-1)^2 = .66$

- Unit 2 weights

- $d^2 = (.92-0)^2 + (.76-0)^2 + (.28-0)^2 + (.12-1)^2 = 2.28$

- Unit 1 wins

- Weights on winning unit are updated

- Giving an updated weight matrix:

$$\begin{aligned} \text{new-unit1-weights} &= [.2 \ .6 \ .5 \ .9] + 0.6([0 \ 0 \ 0 \ 1] - [.2 \ .6 \ .5 \ .9]) = \\ &= [.08 \ .24 \ .20 \ .96] \end{aligned}$$

$$\text{Unit 1: } \begin{bmatrix} .08 & .24 & .20 & .96 \end{bmatrix}$$

$$\text{Unit 2: } \begin{bmatrix} .92 & .76 & .28 & .12 \end{bmatrix}$$

Contoh

Third Weight Update

i1: (1, 1, 0, 0)

i2: (0, 0, 0, 1)

i3: (1, 0, 0, 0)

i4: (0, 0, 1, 1)

- Training sample: i3

$$\begin{array}{l} \text{Unit 1: } \begin{bmatrix} .08 & .24 & .20 & .96 \end{bmatrix} \\ \text{Unit 2: } \begin{bmatrix} .92 & .76 & .28 & .12 \end{bmatrix} \end{array}$$

- Unit 1 weights

- $d^2 = (.08-1)^2 + (.24-0)^2 + (.2-0)^2 + (.96-0)^2 = 1.87$

- Unit 2 weights

- $d^2 = (.92-1)^2 + (.76-0)^2 + (.28-0)^2 + (.12-0)^2 = 0.68$

- Unit 2 wins

- Weights on winning unit are updated

- Giving an updated weight matrix:

$$\begin{aligned} \text{new-unit2-weights} &= [.92 \quad .76 \quad .28 \quad .12] + 0.6([1 \ 0 \ 0 \ 0] - [.92 \quad .76 \quad .28 \quad .12]) = \\ &= [.97 \quad .30 \quad .11 \quad .05] \end{aligned}$$

$$\begin{array}{l} \text{Unit 1: } \begin{bmatrix} .08 & .24 & .20 & .96 \end{bmatrix} \\ \text{Unit 2: } \begin{bmatrix} .97 & .30 & .11 & .05 \end{bmatrix} \end{array}$$

Contoh

Fourth Weight Update

i1: (1, 1, 0, 0)

i2: (0, 0, 0, 1)

i3: (1, 0, 0, 0)

i4: (0, 0, 1, 1)

- Training sample: i4

- Unit 1 weights

- $d^2 = (.08-0)^2 + (.24-0)^2 + (.2-1)^2 + (.96-1)^2 = .71$

- Unit 2 weights

- $d^2 = (.97-0)^2 + (.30-0)^2 + (.11-1)^2 + (.05-1)^2 = 2.74$

- Unit 1 wins

- Weights on winning unit are updated

- Giving an updated weight matrix:

$$\begin{aligned} \text{new-unit1-weights} &= [.08 \ .24 \ .20 \ .96] + 0.6([0 \ 0 \ 1 \ 1] - [.08 \ .24 \ .20 \ .96]) = \\ &= [.03 \ .10 \ .68 \ .98] \end{aligned}$$

$$\text{Unit 1: } \begin{bmatrix} .03 & .10 & .68 & .98 \end{bmatrix}$$

$$\text{Unit 2: } \begin{bmatrix} .97 & .30 & .11 & .05 \end{bmatrix}$$

Contoh

Applying the SOM Algorithm

Data sample utilized

| time (t) | 1 | 2 | 3 | 4 | D(t) | $\eta(t)$ |
|----------|--------|--------|--------|--------|------|-----------|
| 1 | Unit 2 | | | | 0 | 0.6 |
| 2 | | Unit 1 | | | 0 | 0.6 |
| 3 | | | Unit 2 | | 0 | 0.6 |
| 4 | | | | Unit 1 | 0 | 0.6 |

‘winning’ output unit

After many iterations (epochs)
through the data set:

$$\begin{array}{l} \text{Unit 1:} \\ \text{Unit 2:} \end{array} \begin{bmatrix} 0 & 0 & .5 & 1.0 \\ 1.0 & .5 & 0 & 0 \end{bmatrix}$$

Did we get the clustering that we expected?

Contoh

Training samples

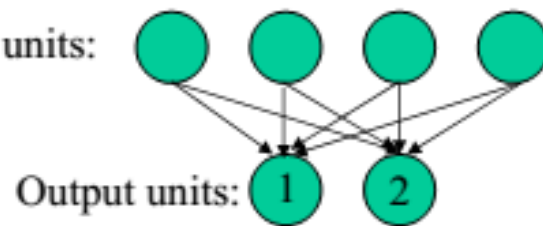
i1: (1, 1, 0, 0)

i2: (0, 0, 0, 1)

i3: (1, 0, 0, 0)

i4: (0, 0, 1, 1)

Input units:



Output units:

Weights

Unit 1: $\begin{bmatrix} 0 & 0 & .5 & 1.0 \end{bmatrix}$

Unit 2: $\begin{bmatrix} 1.0 & .5 & 0 & 0 \end{bmatrix}$

Contoh

Training samples

i1: (1, 1, 0, 0)

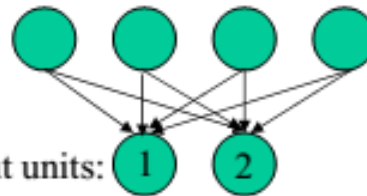
i2: (0, 0, 0, 1)

i3: (1, 0, 0, 0)

i4: (0, 0, 1, 1)

Solution

Input units:



Output units:

Unit 1:

Unit 2:

Weights

$$\begin{bmatrix} 0 & 0 & .5 & 1.0 \\ 1.0 & .5 & 0 & 0 \end{bmatrix}$$

- Sample: i1
 - Distance from unit1 weights
 - $(1-0)^2 + (1-0)^2 + (0-.5)^2 + (0-1.0)^2 = 1+1+.25+1=3.25$
 - Distance from unit2 weights
 - $(1-1)^2 + (1-.5)^2 + (0-0)^2 + (0-0)^2 = 0+.25+0+0=.25$ (winner)
- Sample: i2
 - Distance from unit1 weights
 - $(0-0)^2 + (0-0)^2 + (0-.5)^2 + (1-1.0)^2 = 0+0+.25+0$ (winner)
 - Distance from unit2 weights
 - $(0-1)^2 + (0-.5)^2 + (0-0)^2 + (1-0)^2 = 1+.25+0+1=2.25$

Contoh

Training samples

i1: (1, 1, 0, 0)

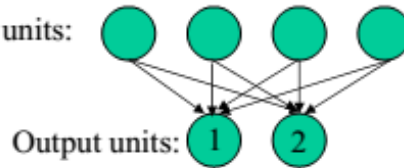
i2: (0, 0, 0, 1)

i3: (1, 0, 0, 0)

i4: (0, 0, 1, 1)

Solution

Input units:



Output units:

Weights

$$\begin{array}{l} \text{Unit 1:} \\ \text{Unit 2:} \end{array} \begin{bmatrix} 0 & 0 & .5 & 1.0 \\ 1.0 & .5 & 0 & 0 \end{bmatrix}$$

- Sample: i3
 - Distance from unit1 weights
 - $(1-0)^2 + (0-0)^2 + (0-.5)^2 + (0-1.0)^2 = 1+0+.25+1=2.25$
 - Distance from unit2 weights
 - $(1-1)^2 + (0-.5)^2 + (0-0)^2 + (0-0)^2 = 0+.25+0+0=.25$ (winner)
- Sample: i4
 - Distance from unit1 weights
 - $(0-0)^2 + (0-0)^2 + (1-.5)^2 + (1-1.0)^2 = 0+0+.25+0$ (winner)
 - Distance from unit2 weights
 - $(0-1)^2 + (0-.5)^2 + (1-0)^2 + (1-0)^2 = 1+.25+1+1=3.25$

Samples i1, i3 cluster with unit 2

Samples i2, i4 cluster with unit 1



Studi Kasus

► Pengenalan Suara yang berfokus pada angka dengan menerapkan MFCC dan SOM

► Pendahuluan

Suara adalah salah satu bentuk alat komunikasi yang cukup efektif. Selain efektif manusia juga sangat familiar dengan komunikasi menggunakan suara. Oleh karena itu banyak berkembang metode pengenalan suara yang dapat diaplikasikan ke dalam komputer sehingga interaksi antara manusia dan komputer dapat dilakukan dengan efektif pula.



Studi Kasus

► Metode Penelitian

Tujuan penelitian ini adalah :

Mengembangkan model pengenalan suara menggunakan MFCC untuk mengekstrak fitur serta algoritma clustering SOM untuk mengenali sinyal suara.



Studi Kasus

► Metode Penelitian

Batasan-batasan dalam penelitian ini adalah :

1. Kata yang digunakan adalah kata dalam bahasa Indonesia untuk angka 1-10, yaitu “Satu”, “Dua”, “Tiga”, “Empat”, “Lima”, “Enam”, “Tujuh”, “Delapan”, “Sembilan”, dan “Sepuluh”.
2. Data yang digunakan merupakan data digital suara, mono, dan direkam dalam lingkungan yang tenang.
3. Data suara dalam format WAV



Studi Kasus

► Metode Penelitian

Batasan-batasan dalam penelitian ini adalah :

4. Data suara yang digunakan berjumlah 20 data suara per satu kata dari 20 orang responden. Pada proses training digunakan 15 data suara dari setiap kata, sedangkan pada proses pengenalan digunakan 5 data suara dari setiap kata.
6. Data suara tidak secara real time



Tugas

- Buat Makalah terkait dengan model pengenalan suara menggunakan MFCC-SOM (format pdf)
- Tugas berkelompok 3 orang 1 kelompok.
- Studi Kasus yang digunakan berbeda antara 1 kelompok dengan kelompok lainnya.
- Tugas dikumpulkan maksimal sebelum UTS di VClass



Selesai