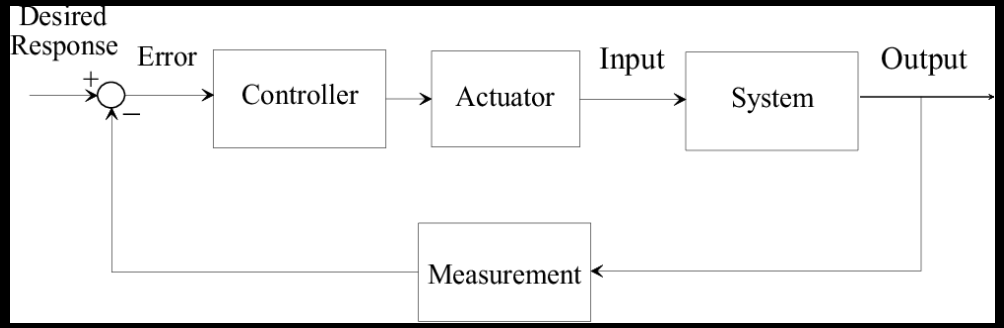
LAPORAN PROYEK JARINGAN SYARAF TIRUAN MENGGUNAKAN ALGORITMA PROPAGASI BELAKANG

Kelompok 1

Ahmad Akbar Habibillah	(1806147804)
George	(1806194883)
Hansel Matthew	(1806194914)
Kemas Muhammad Rizki Fadhila	(1806195072)

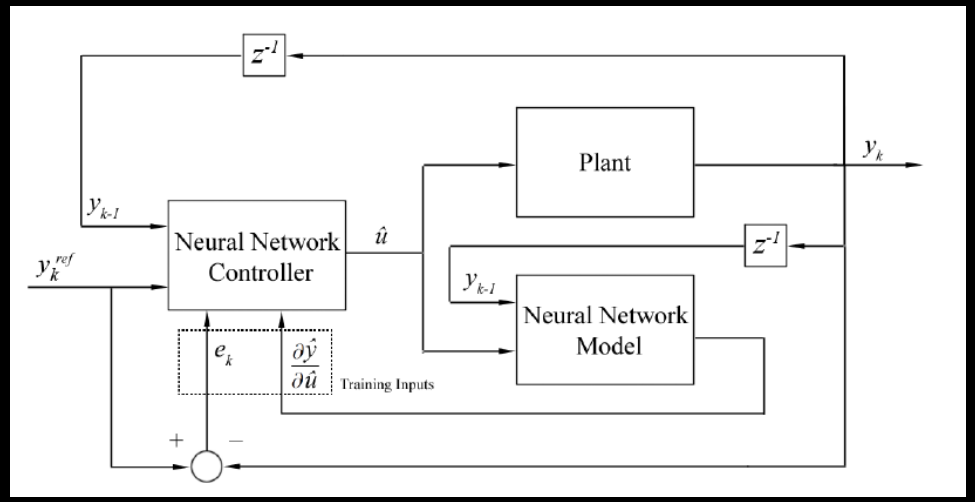
Latar Belakang

Classical Control



- Perlu pemodelan white-box
- Mahal
- Kompleks
- Perlu ditala ulang

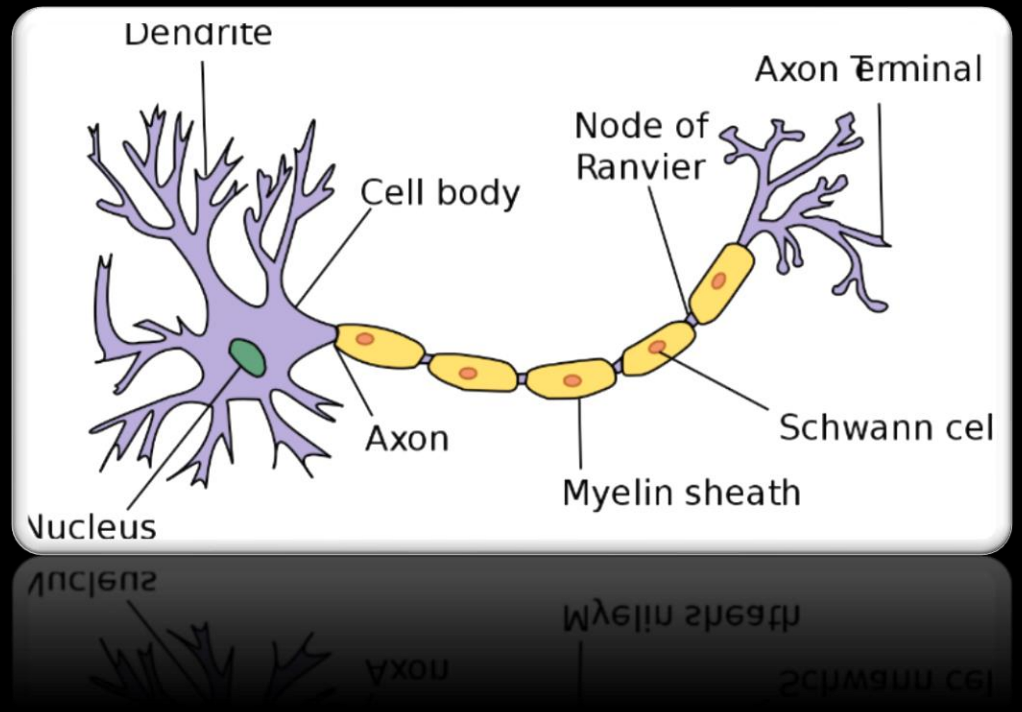
Neural Network Controller



- Tidak perlu dimodelkan, cukup dengan data pasangan input dan output sistem
- Lebih adaptif dan prediktif
- Dapat menala ulang sendiri

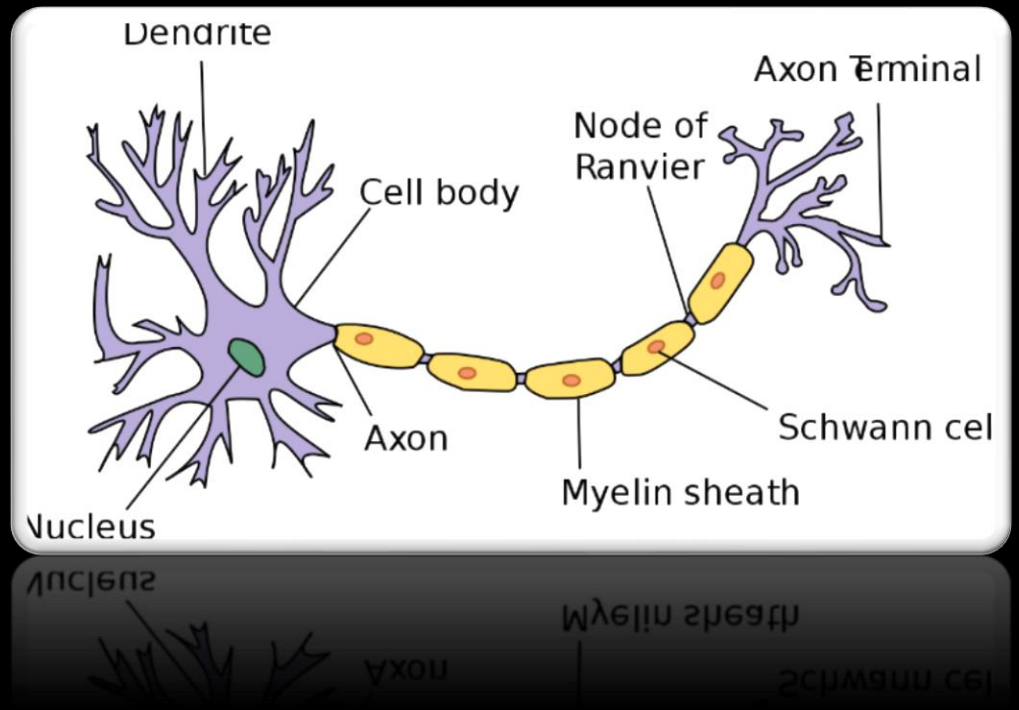
Dasar Teori

ANN merupakan jaringan yang dibuat dengan meniru jaringan syaraf manusia dengan diilhami oleh struktur dan cara kerja otak dan sel syaraf manusia. Neuron otak manusia memiliki 3 komponen penting yaitu Dendrite, Nucleus, dan Axon. Ketiga komponen inilah yang ditiru pada ANN.



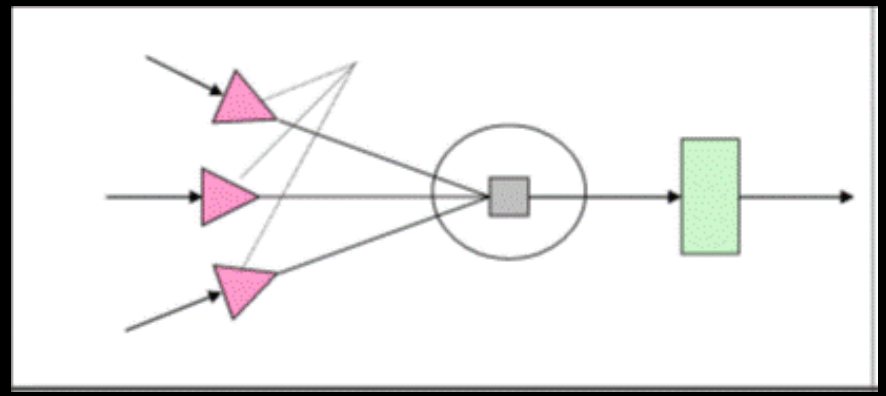
Dasar Teori

Cara kerja dari sebuah neuron adalah akan bereaksi apabila potensial listrik mencapai suatu batasan tertentu. Neuron akan menjumlahkan sinyal yang masuk melalui dendrite yang dikalikan dengan pembobot sinapsis. Proses pembelajaran terjadi dengan perubahan yang terjadi pada sinapsis. Sinyal yang masuk akan dijumlahkan dan dikonversi dengan suatu fungsi aktivitas yang kemudian akan mengeluarkan suatu sinyal pemicu yang dialirkan ke neuron lain.



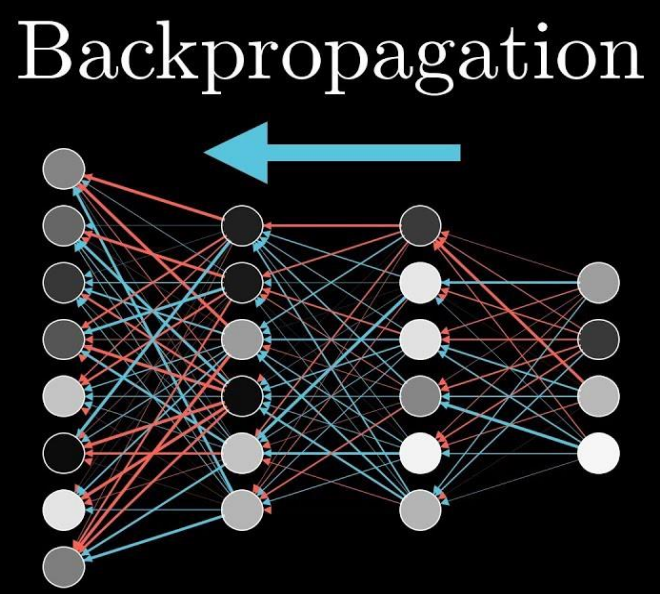
Dasar Teori

Prinsip kerja dari neuron ini kemudian dimodelkan secara matematis. Pemodelan matematika inilah yang menjadi dasar dari ANN. Elemen dasar dari sebuah ANN adalah sebuah neuron. Neuron ini akan mengubah sinyal msukan menjadi sebuah keluaran. Setiap neuron mempunyai suatu inputan yang memiliki bobotnya masing – masing. Sinyal kemudian akan dikonversi menggunakan sebuah fungsi aktivasi. Fungsi aktivasi yang digunakan dapat beragam seperti *Relu, sigmoid, tanh* dan lain lain



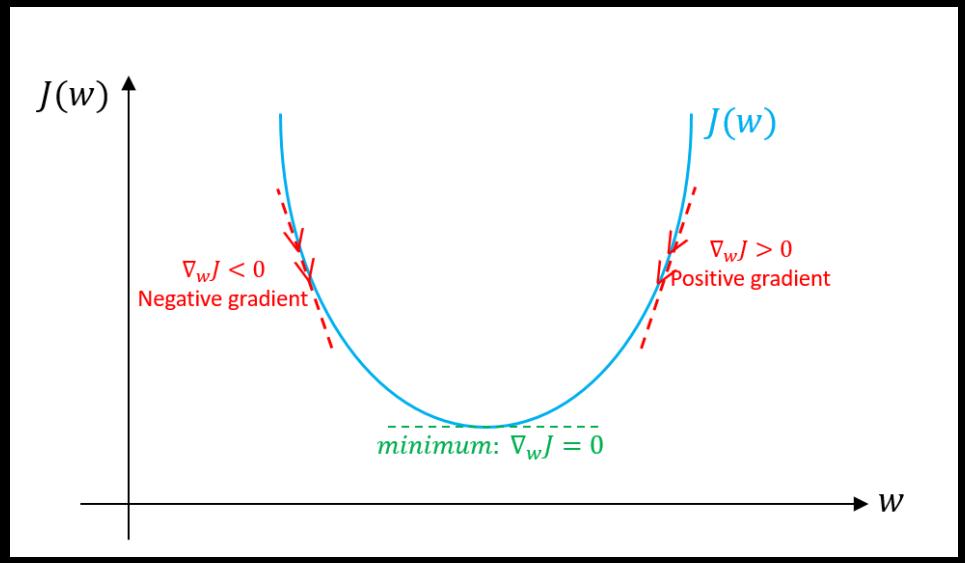
Dasar Teori

Backpropagation adalah suatu algoritma ANN yang memiliki kekuatan utama pada klasifikasi suatu pola atau disebut sebagai *pattern recognition*. Jaringan syaraf *backpropagation* dapat digunakan untuk memprediksi luaran dari sebuah sistem. Pada mulanya data akan diberikan pada sistem. Jika sistem menghasilkan luaran yang tidak sesuai dengan yang diharapkan, maka *backpropagation* akan memodifikasi bobot pada hubungan neuron



Dasar Teori

Gradient Descent adalah algoritma pengoptimalan yang paling umum dalam pembelajaran mesin dan pembelajaran mendalam. Ini adalah algoritma pengoptimalan orde pertama. Hal ini dilakukan dengan menghitung turunan pertama saat melakukan pembaruan pada parameter dimana akan menghasilkan gradien dari fungsi error. Pada setiap iterasi, parameter akan diperbaharui dalam arah yang berlawanan dari gradien fungsi error J(w). Besar kecilnya langkah pada setiap iterasi untuk mencapai minimum lokal ditentukan oleh kecepatan pemelajaran α (Learning Rate).



Mengenai Dataset

Nama : Teaching Assistant Evaluation

Feature :

- Bahasa penutur(biner); 1 = penutur bahasa Inggris, 2 = penutur non-bahasa Inggris
- Instruktur kursus (kategorikal, 25 kategori)
- Kursus (kategorikal, 26 kategori)
- Semester regular atau musim panas (biner) 1 = Musim Panas, 2 = Regular
- Ukuran kelas (numerik)

Class : Low, Medium, High

Jumlah instance : 151

https://archive.ics.uci.edu/ml/datasets/Teaching+Assistant+Evaluation

Langkah Kerja Neural Network + Backpropagation

1. Normalisasi Data
2. Input Data + Data Splitting
3. Forward Pass
4. Kuantitasi Output
5. Backward Pass
6. Pembaharuan Bobot
7. Repeat

1. Normalisasi Data

Data dinormalisasi sehingga tiap feature memiliki range 0 sampai 1 sehingga tidak ada feature yang mendominasi feature lainnya dalam proses gradient descent.

```
%% Normalisasi
feature_norm = zeros(size(feature));
for m = 1 : feature_row
    for n = 1 : feature_col
        feature_norm(m,n) = ((feature(m,n) - min(feature(:,n)))/(max(feature(:,n)) - min(feature(:,n))));
    end
end
```

2. Input Data + Data Splitting

Data dimasukkan ke dalam program dan kemudian dibagi 2 menjadi set Train dan set Test. Pembagian dipisah dengan perbandingan Train:Test sejumlah 70:30.

```
%% Data Input
load('dataset.mat')
feature = dataset(1:105,1:5);
feature = feature{:,:};
class = dataset(1:105,6:8);
class = class{:,:};

[feature_row,feature_col] = size(feature);
[class_row,class_col] = size(class);
```

3. Melakukan proses forward pass

Data dimasukkan ke dalam layer input dan dihitung maju ke hidden layer dan ke output layer menggunakan prinsip neural network.

```
%Forward Pass
%Input -> Hidden
x = feature(n,:);
t = class(n,:);

z_in = bias_xz + x*weight_xz;

for m=1:z_size
    z(1,m) = 1/(1+exp(-z_in(1,m)));
end

%Hidden -> Output
y_in = bias_zy + z*weight_zy;

for l=1:y_size
    y(n,l) = 1/(1+exp(-y_in(1,l)));
end
```

4. Kuantisasi Output

Karena nilai yang ingin dicapai pada nyatanya tidak dapat mencapai nilai 0 atau 1 sempurna, maka dilakukan threshold untuk membulatkan nilai yang mendekati 0 atau 1 ke angka tersebut.

```
%Threshold
for s=1:3
    if y(n,s) >= 0.7
        y(n,s) = 1;
    end
    if y(n,s) <= 0.3
        y(n,s) = 0;
    end
end
```

5. Melakukan proses backward pass

Setelah jaringan telah selesai menghitung dan didapatkan luaran dari jaringan. Luaran tersebut dibandingkan dengan nilai target sesungguhnya dan dihitung sinyal error dan besar koreksi pada tiap bobot jaringan.

```
%Backward Pass
%Output->Hidden
for l=1:y_size
    do_k(1,l) = (y(n,l) - t(1,l)) * (y(n,l)*(1-y(n,l)));
end

delta_zy = (alpha .* z' * do_k);
delta_zy_bias = alpha .* do_k;

%Hidden->Input
sigma_j = do_k * weight_zy';
for m=1:z_size
    do_j(1,m) = (sigma_j(1,m)) .* (z(1,m)*(1-z(1,m)));
end

delta_xz = (alpha .* x' * do_j);
delta_xz_bias = alpha .* do_j;
```

6. Melakukan Pembaharuan Bobot

Bobot diperbaharui sesuai dengan prinsip gradient descent. Metode gradient descent yang digunakan adalah stochastic gradient descent. Dimana bobot akan di update setiap satu buah pasangan pelatihan masuk ke dalam jaringan.

```
%Weight Update
weight_zy = weight_zy - delta_zy - momentum_zy;
weight_xz = weight_xz - delta_xz - momentum_xz;
bias_zy = bias_zy - delta_zy_bias;
bias_xz = bias_xz - delta_xz_bias;
```

6.5 Melakukan Pembaharuan Bobot Menggunakan Momentum

Pada keadaan dimana momentum tidak digunakan, miu akan bernilai 0 sehingga parameter momentum tidak akan berpengaruh terhadap pembaharuan bobot, namun jika momentum digunakan, maka pembaharuan bobot akan berbeda, sehingga momentum akan berpengaruh pada bobot.

```
%Momentum calculation
momentum_zy = miu*delta_zy_old;
momentum_xz = miu*delta_xz_old;
```

7. Repeat!

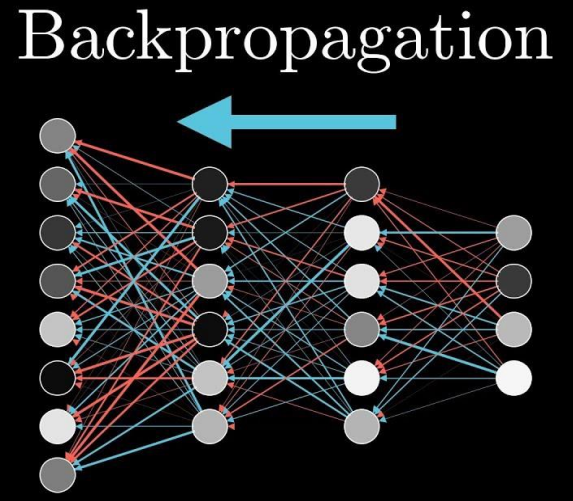
Melakukan step 3-6 sampai kondisi berhenti terpenuhi. Kondisi berhenti dapat berupa epoch telah mencapai epoch maksimal atau error sudah mencapai tingkat yang diinginkan

```
error_per_epoch(1,epoch_count) = sum(error)/feature_row;

disp(epoch_count)
disp(sum(error)/feature_row)

if error_per_epoch(1,epoch_count) < error_target
        stop = 1;
end

epoch_count = epoch_count+1;
```

Variasi Parameter

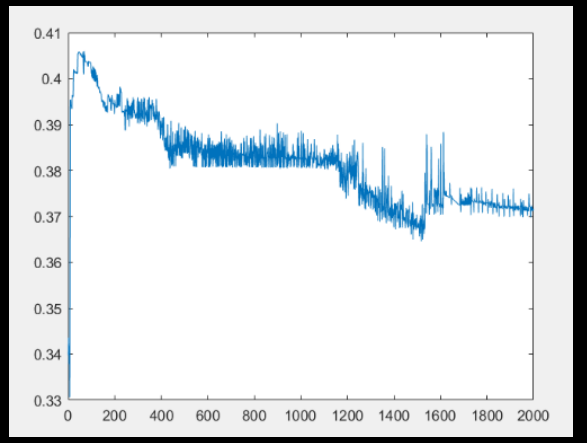
Parameter dari jaringan akan divariasikan yaitu metode inisialisasi , nilai learning rate (α), jumlah neuron pada hidden layer (J), dan menggunakan momentum atau tidak. Variasi dari parameter ini diharapkan dapat memberikan hasil yang dapat menggambarkan pengaruh parameter terhadap kondisi dan hasil dari ANN.

- Learning Rate = 0.1
- Ukuran hidden layer = 7 node
- Target error maksimal = 0.1
- Maximum epoch = 2000 epoch
- Momentum = Tidak Digunakan
- Metode Inisialisasi = Nguyen-Widrow

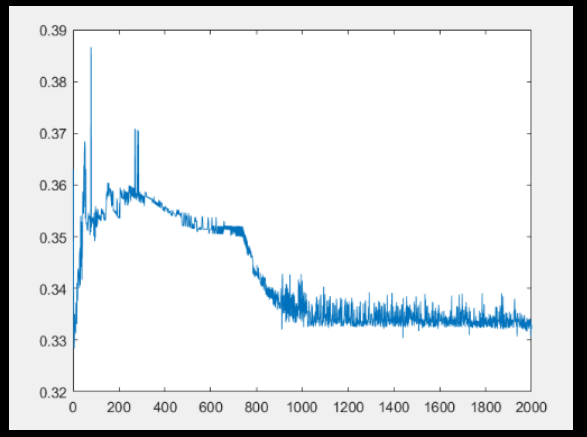


Variasi Metode Inisialisasi - Hansel

Random Initialization
Final Error = 0. 3715
Epoch yang digunakan = 2000
Recognition rate = 39%

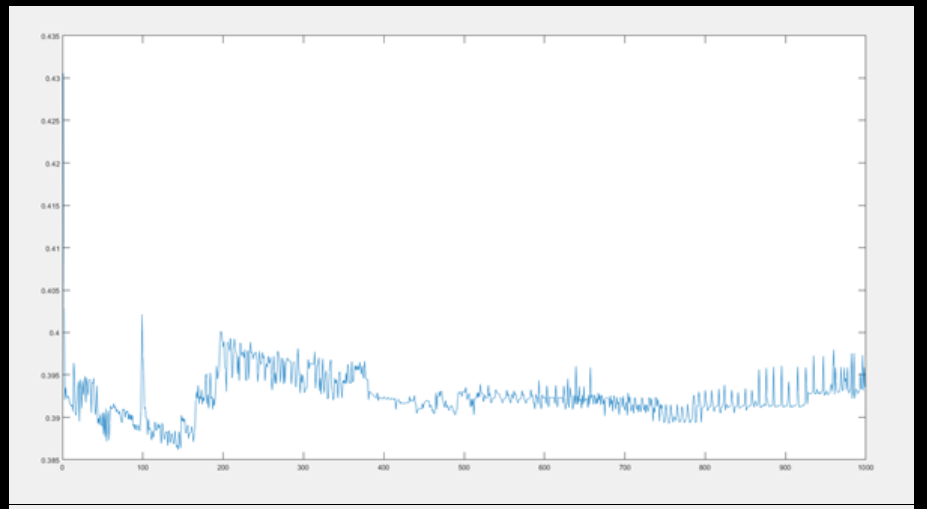


Nguyen-Widrow Initialization
Final Error = 0. 3330
Epoch yang digunakan = 2000
Recognition rate = 39%

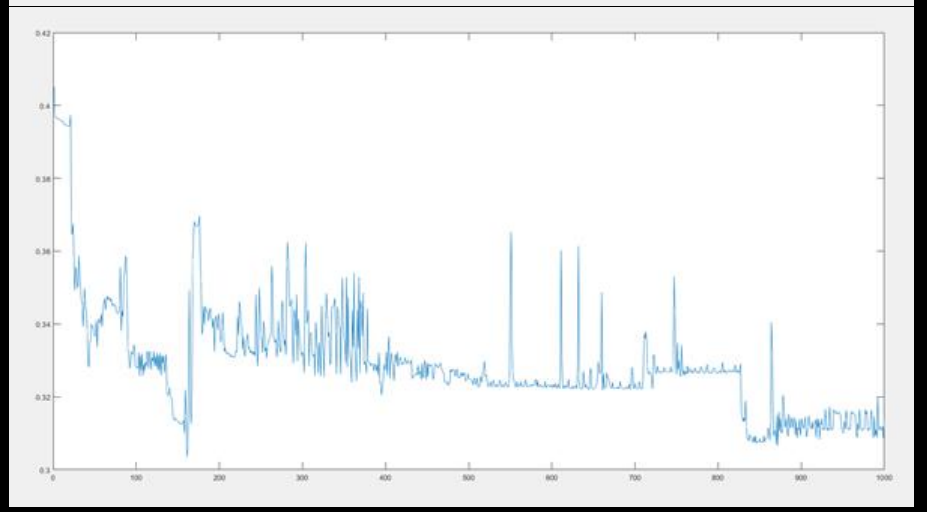


Variasi Metode Inisialisasi – George

Random Initialization
Final Error = 0. 3934
Epoch yang digunakan = 1000
Recognition rate = 32.61%

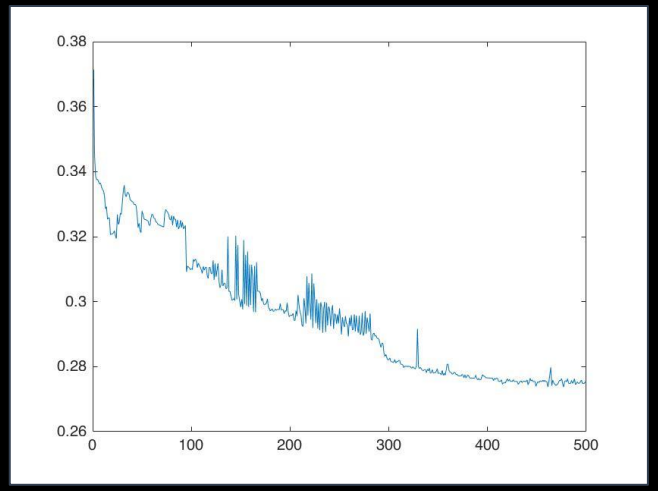


Nguyen-widrow Initialization
Final Error = 0. 3142
Epoch yang digunakan = 1000
Recognition rate = 13.04%

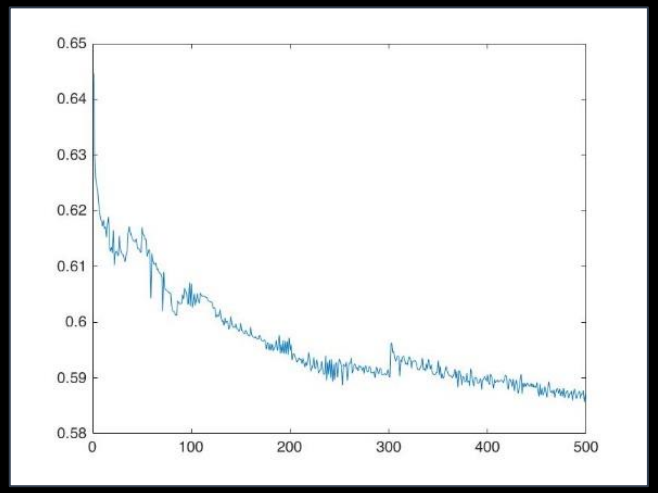


Variasi Metode Inisialisasi – Kemas

Random Initialization
Final Error = 0. 2753
Epoch yang digunakan = 500
Recognition rate = 23,913%

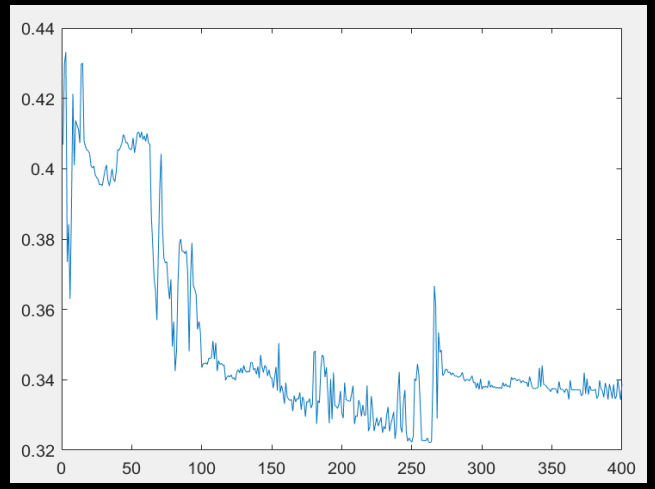


Nguyen-widrow Initialization
Final Error = 0. 5866
Epoch yang digunakan = 500
Recognition rate = 58,696%

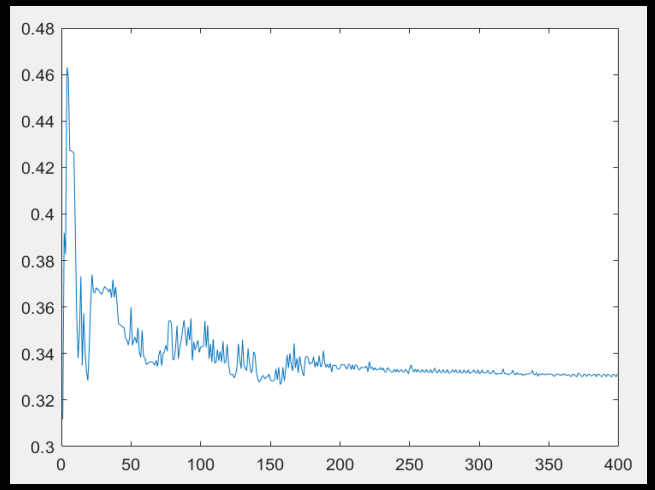


Variasi Metode Inisialisasi – Habib

Random Initialization
Final Error = 0.339
Epoch yang digunakan = 400
Recognition rate = 30%

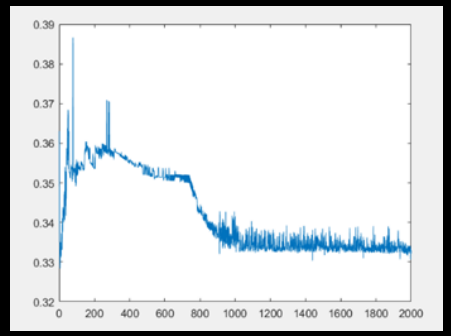


Nguyen-widrow Initialization
Final Error = 0. 3309
Epoch yang digunakan = 400
Recognition rate = 10 %

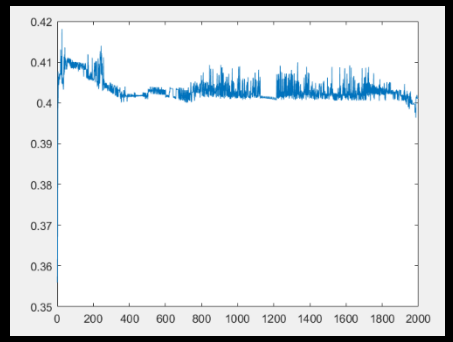


Variasi Learning Rate – Hansel

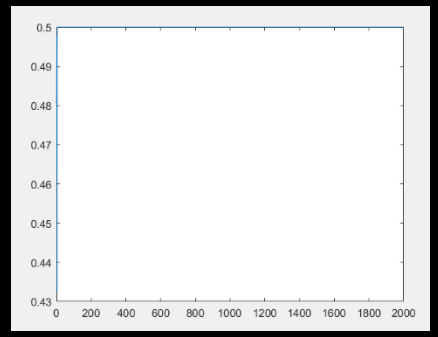
α = 0.1
Final Error = 0. 3330
Epoch yang digunakan =2000
Recognition rate = 39%



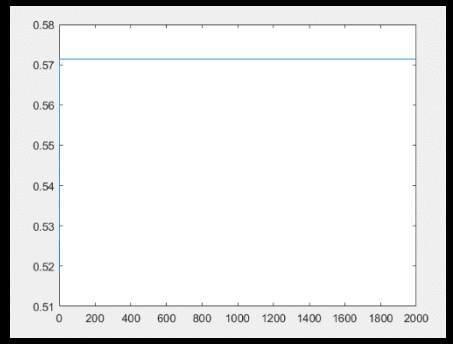
α = 0.2
Final Error = 0. 4286
Epoch yang digunakan = 2000
Recognition rate = 43%



α = 0.4
Final Error = 0. 5000
Epoch yang digunakan = 2000
Recognition rate = 58%

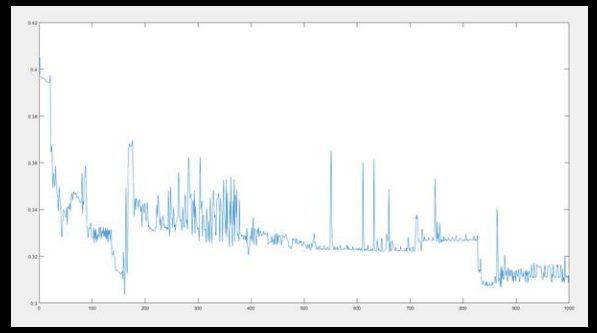


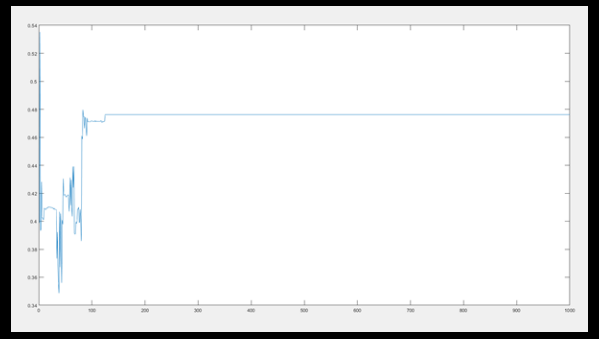
α = 0.8
Final Error = 0. 5714
Epoch yang digunakan = 2000
Recognition rate = 15%



Variasi Learning Rate – George

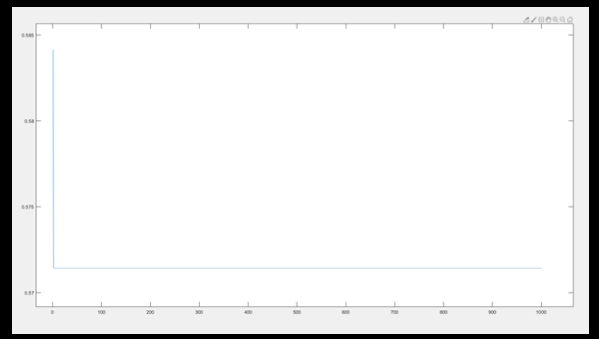
α = 0.1
Final Error = 0. 3142
Epoch yang digunakan = 1000
Recognition rate = 13.04%





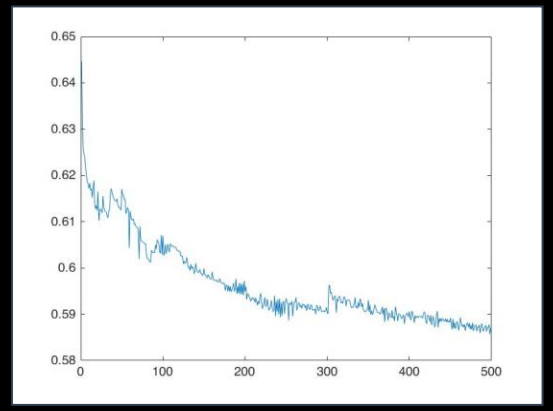
α = 0.2
Final Error = 0. 4762
Epoch yang digunakan = 1000
Recognition rate = 52.17%

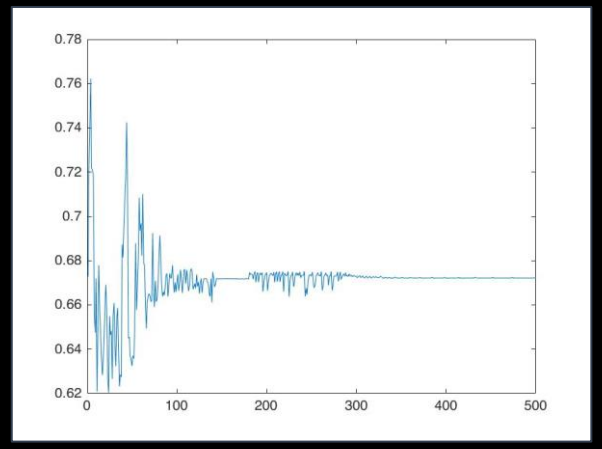
α = 0.3
Final Error = 0. 5714
Epoch yang digunakan = 1000
Recognition rate = 15.22%



Variasi Learning Rate – Kemas

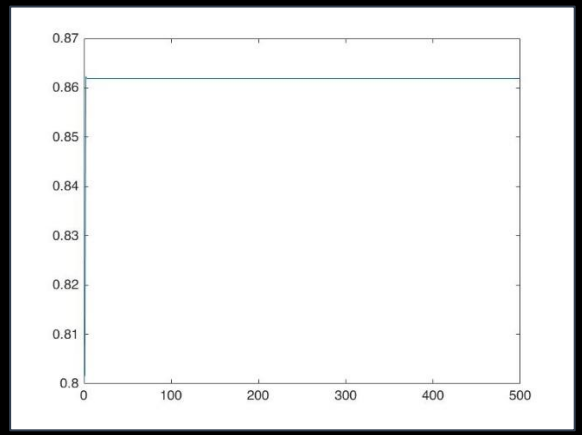
α = 0.05
Final Error = 0. 5866
Epoch yang digunakan = 500
Recognition rate = 58,696%





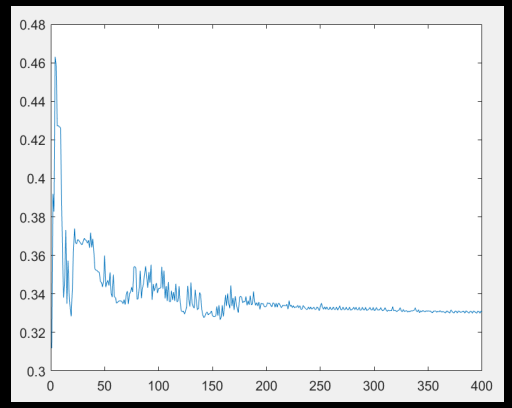
α = 0.2
Final Error = 0. 6722
Epoch yang digunakan = 500
Recognition rate = 58,696%

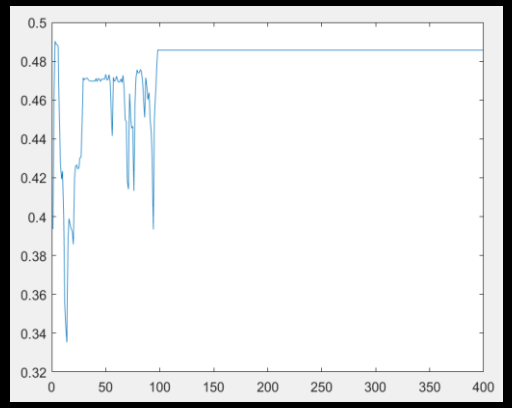
α = 0.5
Final Error = 0. 8619
Epoch yang digunakan = 500
Recognition rate = 58,697%



Variasi Learning Rate – Habib

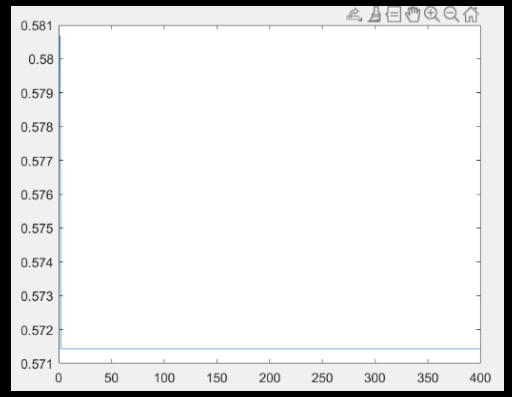
α = 0.1
Final Error = 0. 3309
Epoch yang digunakan = 400
Recognition rate = 10 %

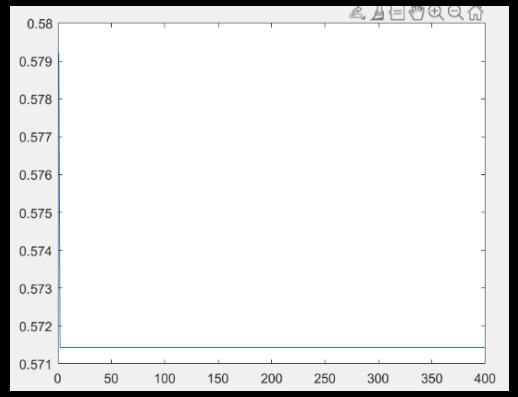




α = 0.2
Final Error = 0. 4857
Epoch yang digunakan = 400
Recognition rate = 54%

α = 0.5
Final Error = 0. 5714
Epoch yang digunakan = 400
Recognition rate = 15%

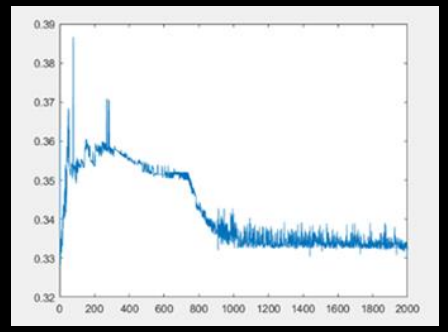


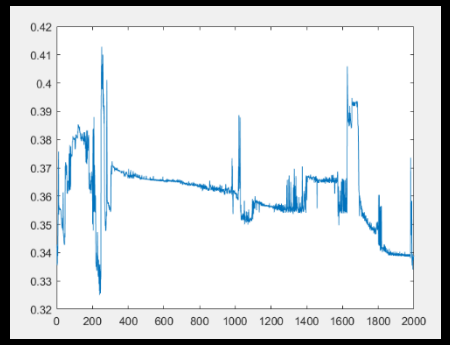


α = 0.7
Final Error = 0. 5714
Epoch yang digunakan = 400
Recognition rate = 15%

Variasi Ukuran Hidden Layer (J) - Hansel

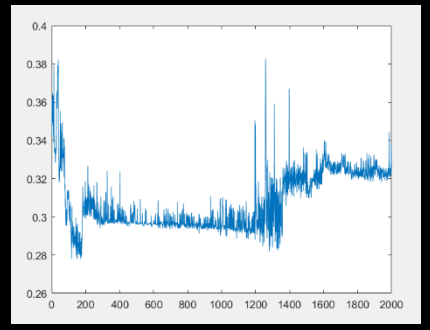
j = 7
Final Error = 0. 3330
Epoch yang digunakan = 2000
Recognition rate = 39%

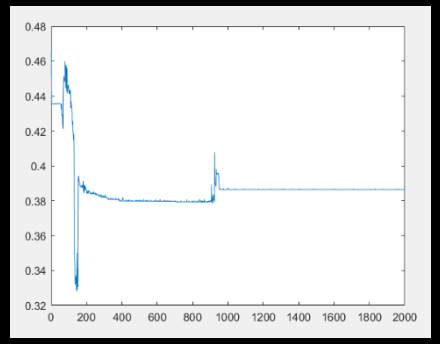




j = 9
Final Error = 0. 3387
Epoch yang digunakan = 2000
Recognition rate = 15%

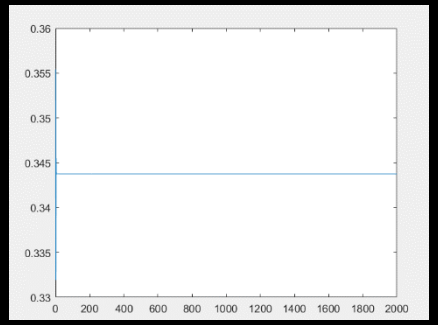
j = 13
Final Error = 0. 3245
Epoch yang digunakan = 2000
Recognition rate = 26%





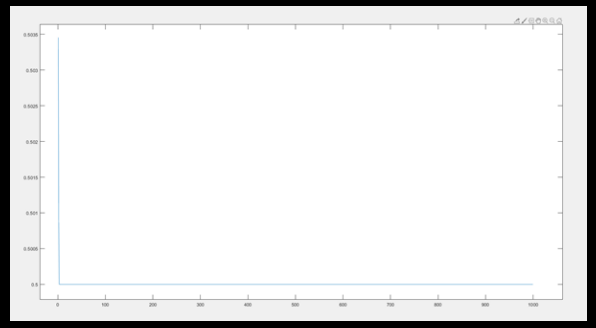
j = 5
Final Error = 0. 3864
Epoch yang digunakan = 2000
Recognition rate = 21%

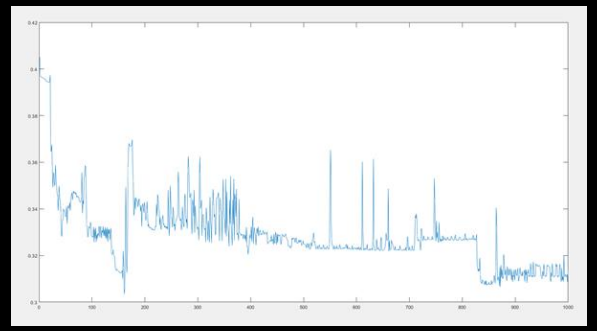
j = 1
Final Error = 0. 3438
Epoch yang digunakan = 2000
Recognition rate = 15%



Variasi Ukuran Hidden Layer (J) – George

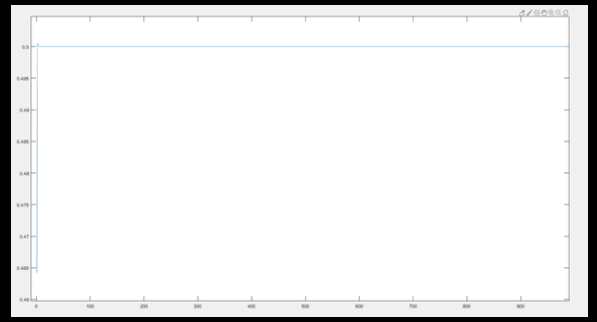
j = 5
Final Error = 0. 5
Epoch yang digunakan = 1000
Recognition rate = 58.69%





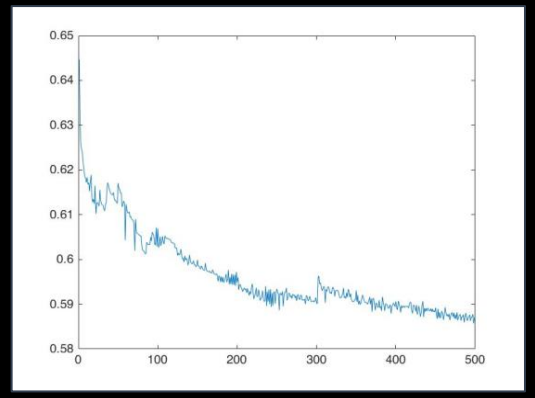
j = 10
Final Error = 0. 3142
Epoch yang digunakan = 1000
Recognition rate = 13.04%

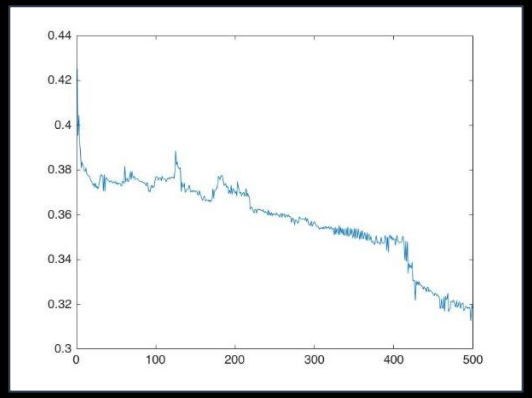
j = 15
Final Error = 0. 5
Epoch yang digunakan = 1000
Recognition rate = 58.69%



Variasi Ukuran Hidden Layer (J) – Kemas

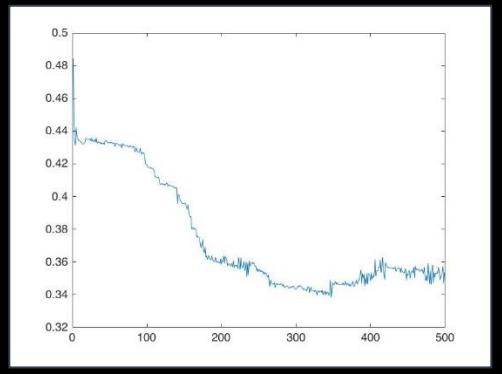
j = 6
Final Error = 0. 5866
Epoch yang digunakan = 500
Recognition rate = 58.696%





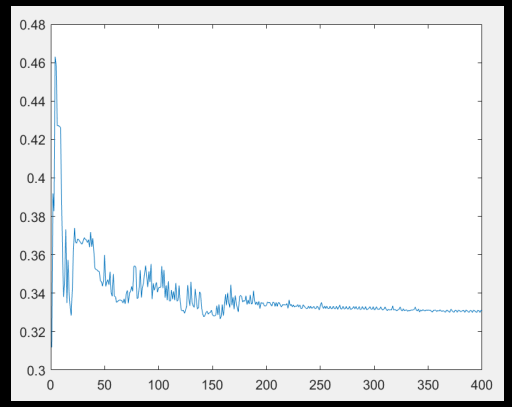
J = 16
Final Error = 0. 3175
Epoch yang digunakan = 500
Recognition rate = 47,826%

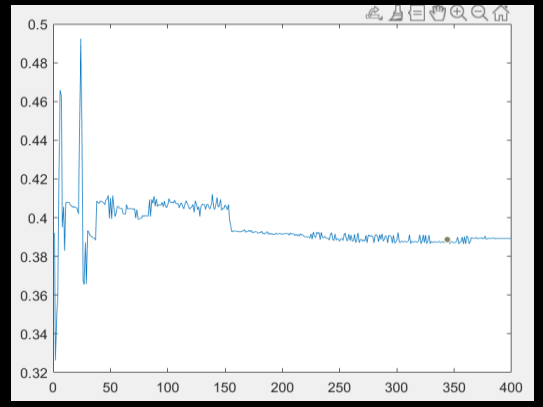
j = 26
Final Error = 0. 3525
Epoch yang digunakan = 500
Recognition rate = 34,783%



Variasi Ukuran Hidden Layer (J) – Habib

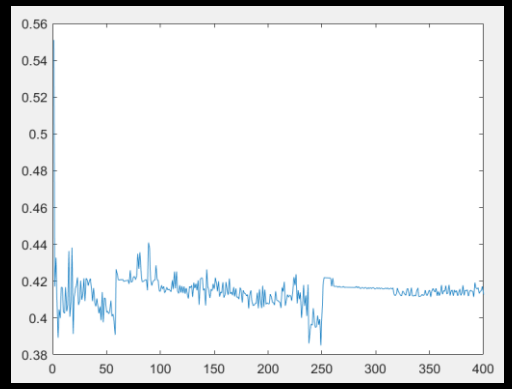
j = 7
Final Error = 0. 3309
Epoch yang digunakan = 400
Recognition rate = 10 %

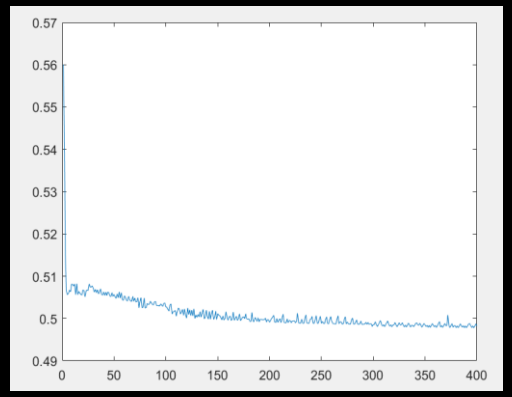




J = 10
Final Error = 0. 3892
Epoch yang digunakan = 400
Recognition rate = 32%

j = 15
Final Error = 0. 4126
Epoch yang digunakan = 400
Recognition rate = 41%

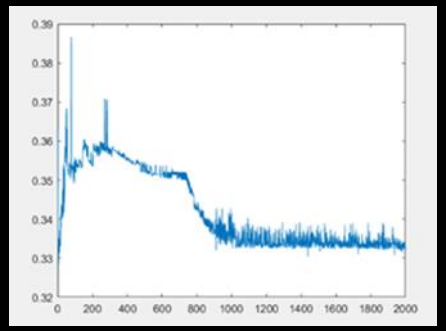


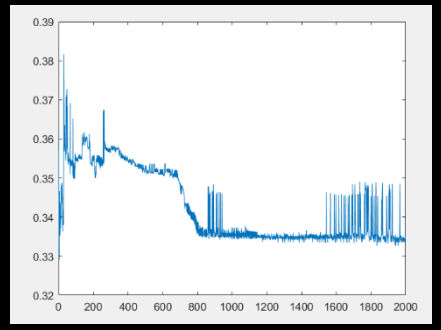


J = 20
Final Error = 0. 4982
Epoch yang digunakan = 400
Recognition rate = 17%

Variasi Nilai Koefisien Momentum (μ) - Hansel

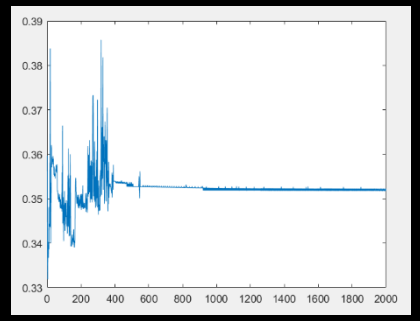
μ = 0
Final Error = 0. 3330
Epoch yang digunakan = 2000
Recognition rate = 39%

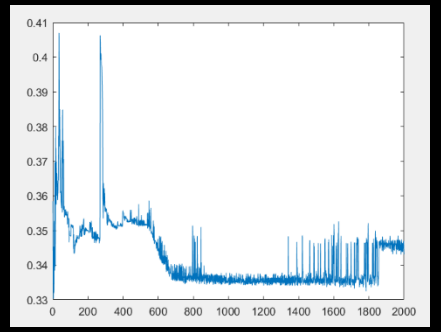




μ = 0.1
Final Error = 0. 3352
Epoch yang digunakan = 2000
Recognition rate = 41%

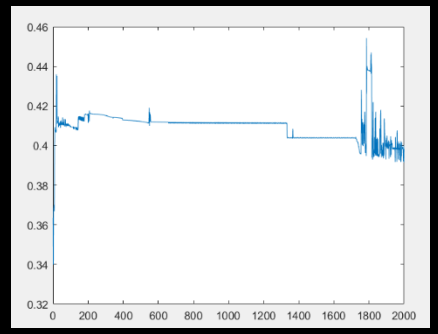
μ = 0.2
Final Error = 0. 3518
Epoch yang digunakan = 2000
Recognition rate = 39%





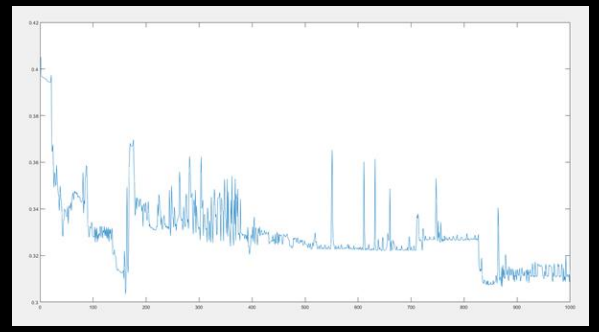
μ = 0.4
Final Error = 0. 4524
Epoch yang digunakan = 2000
Recognition rate = 54%

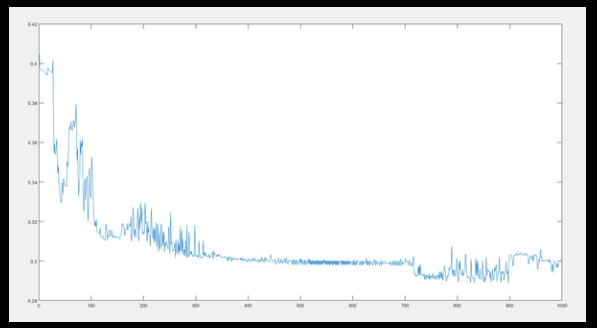
μ = 0.8
Final Error = 0. 4524
Epoch yang digunakan = 2000
Recognition rate = 54%



Variasi Nilai Koefisien Momentum (μ) – George

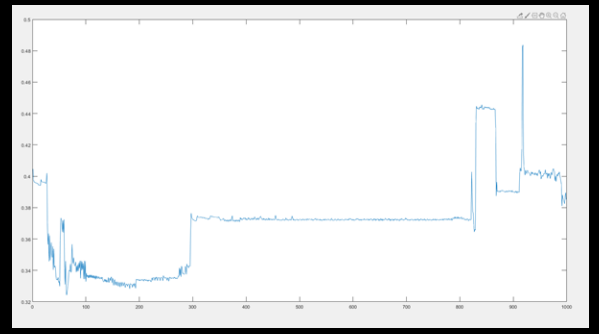
μ = 0
Final Error = 0. 3142
Epoch yang digunakan = 1000
Recognition rate = 13.04%

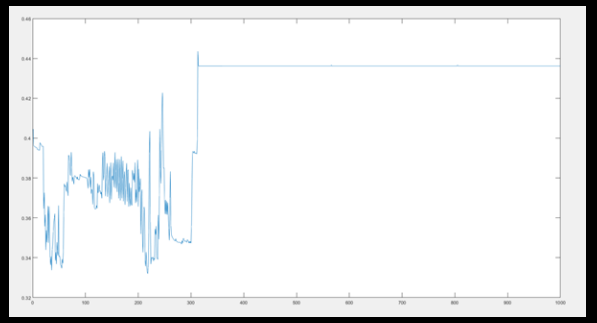




μ = 0.05
Final Error = 0. 2999
Epoch yang digunakan = 1000
Recognition rate = 21.74%

μ = 0.1
Final Error = 0. 3851
Epoch yang digunakan = 1000
Recognition rate = 26.08%

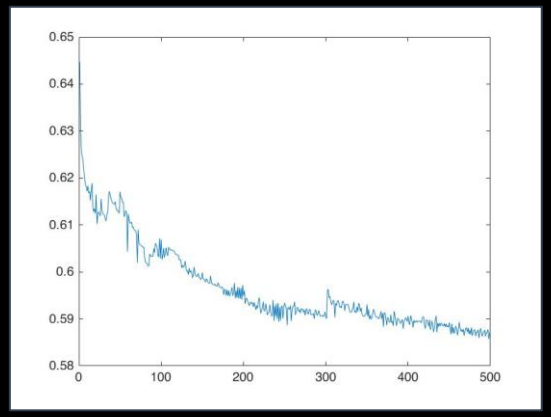




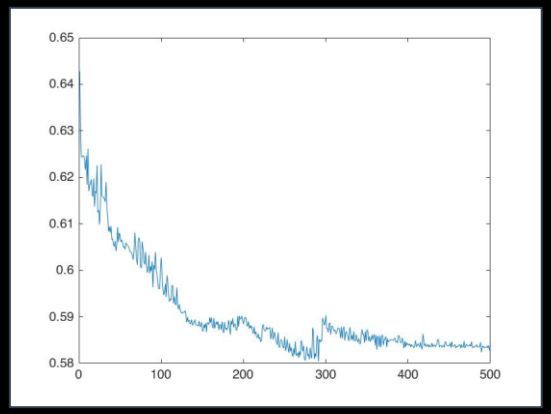
μ = 0.2
Final Error = 0. 4362
Epoch yang digunakan = 1000
Recognition rate = 26.08%

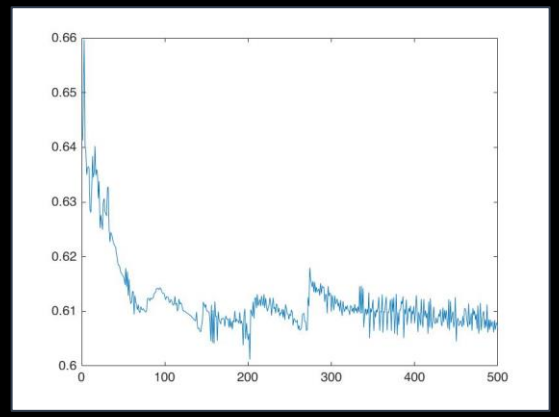
Variasi Nilai Koefisien Momentum (μ) – Kemas

μ = 0
Final Error = 0. 5866
Epoch yang digunakan = 500
Recognition rate = 58,696%



μ = 0.3
Final Error = 0. 5838
Epoch yang digunakan = 500
Recognition rate = 58,697%

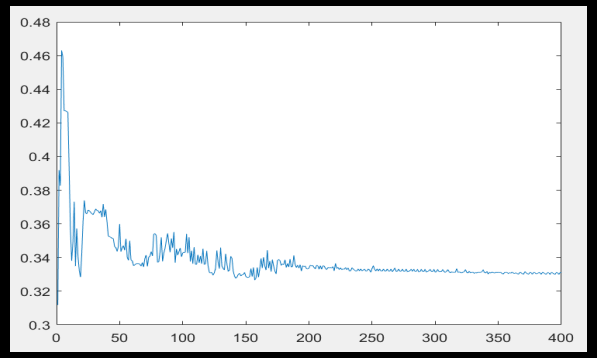


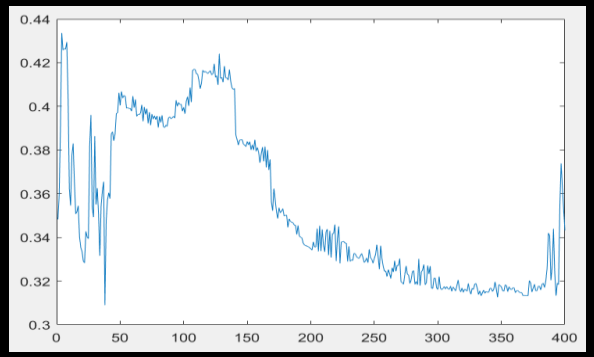


μ = 0.6
Final Error = 0. 6066
Epoch yang digunakan = 500
Recognition rate = 58,696%

Variasi Nilai Koefisien Momentum (μ) – Habib

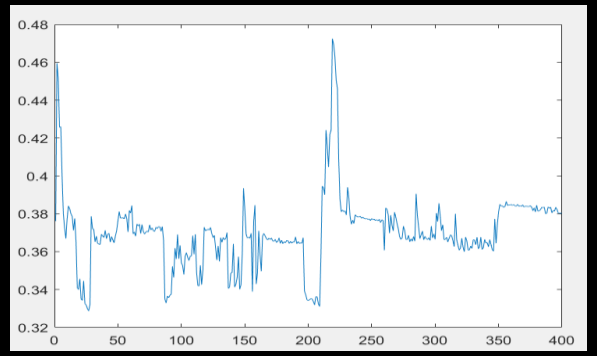
μ = 0
Final Error = 0. 3309
Epoch yang digunakan = 400
Recognition rate = 10%

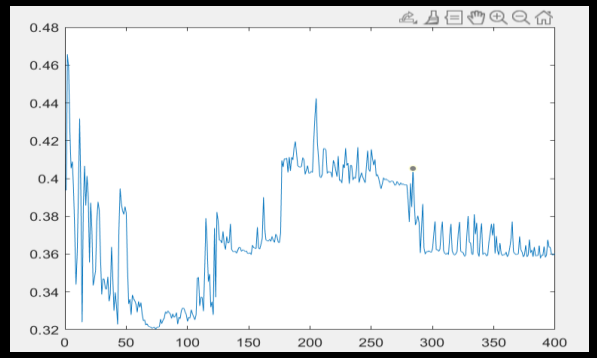




μ = 0.1
Final Error = 0. 3432
Epoch yang digunakan = 400
Recognition rate = 32%

μ = 0.2
Final Error = 0. 3803
Epoch yang digunakan = 400
Recognition rate = 23%





μ = 0.3
Final Error = 0. 3603
Epoch yang digunakan = 400
Recognition rate = 23%

KESIMPULAN

1. *Backpropagation* merupakan metode ANN yang merupakan metode pembelajaran sistem yang mengubah nilai bobot dari neuron relatif terhadap error yang diperoleh
2. Inisialisasi nguyen-widrow mempercepat ANN untuk mencapai konvergensi
3. Nilai learning rate (α) berpengaruh terhadap durasi ANN untuk mencapai konvergensi dan potensi untuk mencapai konvergensi tersebut.
4. Jumlah neuron hidden layer berpengaruh terhadap hasil luaran ANN dimana semakin banyak jumlah neuron akan memberikan hasil yang lebih baik namun perlu diwaspadai pengaruh dari jumlah neuron yang terlalu banyak
5. Penggunaan momentum akan mengurangi noise pergerakan ANN dan mempercepat ANN untuk mencapai konvergensi

nkyou 谢谢 Merci Terima Kasih ありがとう Danke Gra

Obrigada Matur Nuwun ขอบคุณ Спасибо 감사합니다

Gratia Hatur Nuhun شکرا cam ơn Bedankt Ви благодаро

Hvala vam Շնորհակալություն Rahmat Takk skal du h

nkyou 谢谢 Merci **Terima Kasih** ありがとう Danke Gr

Obrigada Matur Nuwun ขอบคุณ Спасибо 감사합니

Gratia Hatur Nuhun شکرا cam ơn Bedankt Ви благодаро

Hvala vam Շնորհակալություն Rahmat Takk skal du h