Laporan

Penerapan Algoritma JST MLP Pelatihan Propagasi Balik Menggunakan Software MatLab

Disusun untuk memenuhi tugas pada mata kuliah Kecerdasan Buatan Bab Learning

Oleh

Ufra Neshia 1107124177  
Ivan Rekyan Fitrayana 1107090021  
Adyo Subhodo 1107100014



Program Studi Ilmu Komputasi

Fakultas Teknik

Telkom University

2014

**Data Training:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tes A** | **Tes B** | **Tes C** | **Tes D** | **target (1: Kelas A, 0: Kelas B)** |
| 0.79 | 0.87 | 0.25 | 0.12 | 0 |
| 0.78 | 0.26 | 0.35 | 0.43 | 0 |
| 0.34 | 0.21 | 0.84 | 0.41 | 0 |
| 0.8 | 0.9 | 1 | 0.78 | 1 |
| 0.88 | 1 | 0.86 | 0.2 | 1 |
| 0.34 | 0.24 | 0.86 | 0.92 | 1 |
| 0.77 | 0.22 | 0.24 | 0.78 | 1 |
| 0.33 | 0.89 | 0.26 | 0.76 | 1 |
| 0.12 | 0.68 | 0.17 | 0.25 | 1 |
| 0.27 | 0.23 | 0.33 | 0.22 | 1 |

**Data Uji:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tes A** | **Tes B** | **Tes C** | **Tes D** | **target (1: Kelas A, 0: Kelas B)** |
| 0.67 | 0.35 | 0.79 | 0.44 | 0 |
| 0.23 | 0.78 | 0.77 | 0.32 | 0 |
| 0.35 | 0.45 | 0.25 | 0.85 | 0 |
| 0.89 | 0.9 | 0.32 | 0.67 | 1 |
| 0.1 | 0.8 | 0.8 | 0.67 | 1 |
| 0.15 | 0.23 | 0.68 | 0.74 | 1 |

1. Merancang Arsitektur JST yang Diperlukan

Metode Jaringan Syaraf Tiruan yang akan dipergunakan adalah Multilayer Perception, yaitu terdiri atas layer input, layer output dan minimal satu layer tersembunyi (hidden layer). Sedang algoritma pelatihan yang akan dipergunakan adalah Algoritma Pelatihan Propagasi Balik (Backpropagation Algorithm), di mana dilakukan dua langkah perhitungan; perhitungan maju 🡪 menghitung galat (error) antara keluaran aktual dengan target serta perhitungan mundur 🡪 menggunakan galat yang didapat untuk memperbaiki bobot-bobot sinaptik semua neuron.

Selanjutnya adalah menentukan jumlah neuron di setiap layer:

1. Jumlah neuron layer input 🡪 4 neuron (karena ada 4 tes)
2. Jumlah neuron layer output 🡪 1 neuron
3. Jumlah Neuron hidden layer 🡪 tidak ada rumus baku, karena itu digunakan rumus pendekatan , sehingga , didapati jumlah neuron hidden layer adalah 2 neuron.

* Karena jumlah data dan neuron hidden layer yang sedikit, maka hanya ada 1 hidden layer sehingga rancangan dasar arsitektur JST soal ini adalah 4-2-1 (4 Neuron Input, 2 Neuron Hidden Layer, 1 Neuron Output layer)

1. Menentukan LR, maxEpoch dan MaxErrorKuadrat: pada dasarnya ditentukan dengan trial dan error, karena itu kami menginisialisasinya dengan:

LR=0.005, maxEpoch=3500, maxError=0.0001.

1. Implementasi Rancangan JST dalam Bentuk Kode Program MatLab
2. Kode Program Training dan Screenshot Hasilnya

clc

clear

Input = xlsread('Data\_1.xlsx') % Data Trainning Input

Output = xlsread('Output\_1.xlsx') % Data Trainning Output

JumData = length(Input(:,1)); % Jumlah data trainning (banyak baris data)

JumInput = length(Input(1,:)); % Jumlah data input (banyak kolom data)

JOneuron = length(Output(1,:)); % Jumlah neurons pada Output layer

JHneuron = 2; % Jumlah neurons pada Hidden layer

LR = 0.005; % Learning Rate

Epoch = 3500; % Maksimum iterasi

MaxMSE = 0.0001; % Maksimum MSE

W1 = [];

for ii=1:JHneuron,

W1 = [W1 ; (rand(1,JumInput)\*2-1)];

end

W1 = W1';

W2 = [];

for jj=1:JOneuron,

W2 = [W2 ; (rand(1,JHneuron)\*2-1)];

end

W2 = W2';

MSEepoch = MaxMSE + 1; % MSE untuk 1 epoch

MSE = []; % List MSE untuk seluruh epoch

ee = 1; % Index Epoch

while (ee <= Epoch) & (MSEepoch > MaxMSE)

MSEepoch = 0;

if (ee>1000), LR=0.004; end

for i=1:JumData,

CP = Input(i,:); % Current Input

CT = Output(i,:); % Current Output

%-----------------------------------------------------------

% Propagasi Maju untuk mendapatkan Output, Error, dan MSE

% Fungsi aktivasi yang digunakan : sigmoid

%-----------------------------------------------------------

A1=[];

for ii=1:JHneuron,

v = CP\*W1(:,ii);

A1 = [A1 1/(1+exp(-v))];

end

A2=[];

for jj=1:JOneuron,

v = A1\*W2(:,jj);

A2 = [A2 1/(1+exp(-v))];

end

Error = CT - A2;

for kk=1:length(Error),

MSEepoch = MSEepoch + Error(kk)^2;

end

%-----------------------------------------------------------

% Propagasi Mundur untuk update Bobot (W1 dan W2)

%-----------------------------------------------------------

for kk=1:JOneuron,

D2(kk) = A2(kk) \* (1-A2(kk)) \* Error(kk);

end

dW2 = [];

for jj=1:JHneuron,

for kk=1:JOneuron,

delta2(kk) = LR \* D2(kk) \* A1(jj);

end

dW2 = [dW2 ; delta2];

end

for jj=1:JHneuron,

D1(jj) = A1 \* (1-A1)' \* D2 \* W2(jj,:)';

end

dW1 = [];

for ii=1:JumInput,

for jj=1:JHneuron,

delta1(jj) = LR \* D1(jj) \* CP(ii);

end

dW1 = [dW1 ; delta1];

end

W1 = W1 + dW1; % W1 baru

W2 = W2 + dW2; % W2 baru

end

MSE = [MSE (MSEepoch/JumData)];

ee = ee + 1;

end

plot(MSE);

xlabel('Epoch')

ylabel('MSE')

display(MSE);

display(W1);

display(W2);

save AI.mat W1 W2 MSE JHneuron JOneuron LR

Bobot W1 dan W2 hasil iterasi program:

W1 =

-0.0745 -0.3899

-1.1590 0.0660

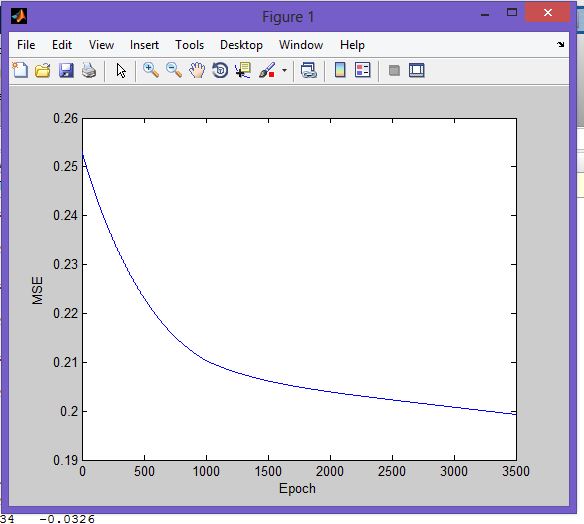
-0.7834 -0.0326

0.1621 1.2154

W2 =

-0.2140

1.5231



Visualisasi perubahan nilai error selama 3500 iterasi

1. Kode Program Uji dan Screenshoot hasilnya

clc

clear

%--------------------------------------------------------------------------

% Load Bobot hasil pelatihan dari file Bobot\_Trainning.mat

%--------------------------------------------------------------------------

load AI.mat

InputTest = xlsread('Data\_2.xlsx') % Data Testing Input

OutputTest = xlsread('Output\_2.xlsx') % Data Testing Output

JumData = length(InputTest(:,1));

JumBenar = 0;

JumSalah = 0;

for i=1:JumData,

CP = InputTest(i,:);

A1 = [];

for ii=1:JHneuron,

v = CP\*W1(:,ii);

A1 = [A1 1/(1+exp(-v))];

end

A2 = [];

for jj=1:JOneuron,

v = A1\*W2(:,jj);

A2 = [A2 1/(1+exp(-v))];

end

%----------------------------------------

% Pemetaan A2 menjadi kelas keputusan

% Jika A2 < 0.5 maka Kelas = 0

% Jika A2 > 0.5 maka Kelas = 1

%----------------------------------------

for jj=1:JOneuron,

if A2(jj) < 0.5,

Kelas = 0;

else

Kelas = 1;

end

end

if Kelas==OutputTest(i),

JumBenar = JumBenar + 1;

else

JumSalah = JumSalah + 1;

end

end

display(JumBenar);

display(JumSalah);

display(['Akurasi JST = ' num2str(JumBenar/JumData)]);

InputTest1 = xlsread('Data\_1.xlsx') % Data Testing Input

OutputTest1 = xlsread('Output\_1.xlsx') % Data Testing Output

JumData = length(InputTest1(:,1));

JumBenar = 0;

JumSalah = 0;

for i=1:JumData,

CP = InputTest1(i,:);

A1 = [];

for ii=1:JHneuron,

v = CP\*W1(:,ii);

A1 = [A1 1/(1+exp(-v))];

end

A2 = [];

for jj=1:JOneuron,

v = A1\*W2(:,jj);

A2 = [A2 1/(1+exp(-v))];

end

for jj=1:JOneuron,

if A2(jj) < 0.5,

Kelas = 0;

else

Kelas = 1;

end

end

if Kelas==OutputTest1(i),

JumBenar = JumBenar + 1;

else

JumSalah = JumSalah + 1;

end

end

display(JumBenar);

display(JumSalah);

display(['Akurasi JST dengan data training= ' num2str(JumBenar/JumData)]);

Hasil pengujian bobot dengan program uji menggunakan data uji dan data training:

1. Dengan data Uji 🡪 total data 6, 3 salah dan 3 benar. Tingkat kesalahan 50%
2. Dengan data Training, total data 10, 7 benar dan 3 salah. Tingkat kesalahan: 30%
3. Analisis dan Kesimpulan

Dari hasil penerapan algoritma JST pada kode program, didapati beberapa factor yang menentukan tingkat keakuratan program: learning rate, jumlah epoch maksimal, dan tingkat error minimal. Semakin kecil learning rate maka penurunan error juga melambat namun mengurangi tingkat kesalahan, karena LR yang terlalu besar dapat menjadikan grafik error tidak stabil.

Jumlah epoch dan error berperan untuk menghentikan program. Semakin kecil epoch maksimal dan/atau semakin besar tingkat error yang diinginkan maka program akan lebih sedikit melakukan perhitungan (makin cepat selesai).

Dalam kasus ini entah kenapa perubahan epoch yang signifikan (dari puluhan jadi ratusan) tidak mampu mengubah hasil pengetesan, padahal teorinya semakin besar Epoch/semakin kecil error maka akan makin akurat pula hasil tesnya.