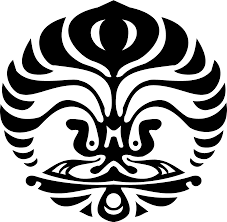
**LAPORAN PROYEK**

**SISTEM BERBASIS PENGETAHUAN**

**“IDENTIFIKASI SISTEM MENGGUNAKAN NEURAL NETWORK”**



Dibuat oleh :

Nama : Suwandi Dwi Sahputro

NPM : 1306413605

Dosen: Prof. Dr. Eng. Drs. Beyamin Kusumoputro, M.Eng.

**DEPARTEMEN TEKNIK ELEKTRO**

**FAKULTAS TEKNIK UNIVERSITAS INDONESIA**

**MARET 2015**

**DAFTAR ISI**

[**BAB 1. PENDAHULUAN** 1](#_Toc449681865)

[**1.1 LATAR BELAKANG NEURAL NETWORK** 1](#_Toc449681866)

[**1.2 ARSITEKTUR NEURAL NETWORK** 2](#_Toc449681867)

[**1.3 METODE PEMBELAJARAN** 3](#_Toc449681868)

[**1.4 IDENTIFIKASI SISTEM MENGGUNAKAN NEURAL NETWORK** 4](#_Toc449681869)

[**BAB 2. BACKPROPAGATION** 5](#_Toc449681870)

[**2.1 ALGORITMA BACKPROPAGATION** 5](#_Toc449681871)

[**2.2 FAKTOR-FAKTOR DALAM PEMBELAJARAN** 8](#_Toc449681872)

[**BAB 3. PERCOBAAN** 9](#_Toc449681873)

[**3.1 VARIASI PERCOBAAN** 9](#_Toc449681874)

[**3.2 PARAMETER KEBERHASILAN PERCOBAAN** 9](#_Toc449681875)

[**3.3 PROSEDUR PERCOBAAN** 9](#_Toc449681876)

[**3.4 PROSES PENGOLAHAN DATA INPUT** 10](#_Toc449681877)

[**3.5 METODE PERCOBAAN** 10](#_Toc449681878)

[**BAB 4. HASIL PERCOBAAN** 11](#_Toc449681879)

[**BAB 5. ANALISA PERCOBAAN** 14](#_Toc449681880)

[**5.1 ANALISA GRAFIK** 14](#_Toc449681881)

[**BAB 6. KESIMPULAN** 15](#_Toc449681882)

[**REFERENSI** 16](#_Toc449681883)

**DAFTAR GAMBAR**

[Gambar 1.1.1 Jaringan Saraf Manusia 1](#_Toc415845167)

[Gambar 1.1.2 Struktur Sederhana ANN 2](#_Toc415845168)

Gambar 1.2.1 Skema Neural Network...................................................................................... 2

Gambar 1.4.1 Arsitektur Neural Network untuk Sistem Identifikasi....................................... 4

Gambar 4.1 Grafik MSSE pada tiap epoch dengan α=0.5, momentum=0.5 dan hidden layer=15....................................................................................................................................11

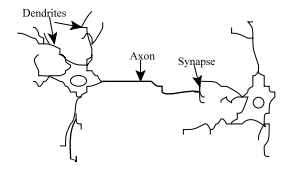
Gambar 4.2 Grafik perbandingan output hasil training dengan target.....................................12

Gambar 4.2 Grafik perbandingan output hasil testinng dengan target.....................................13

# **BAB 1. PENDAHULUAN**

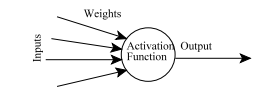
## **1.1 LATAR BELAKANG NEURAL NETWORK**

Artificial Neural Network (ANN) merupakan sebuah sistem yang mengadopsi cara kerja sistem jaringan saraf pada makhluk hidup. Jaringan saraf pada makhluk hidup memiliki kemampuan menerima impuls yang random dari sensor pada tubuh kemudian impuls tersebut masuk ke dalam jaringan saraf lalu disampaikan ke otak dan dikembalikan hasil keluarannya sebagai respon. Jaringan saraf makhluk hidup terdiri dari banyak sel. Ketika impuls masuk ke jaringan saraf, kemudian impuls tersebut disalurkan ke dalam neuron yang lain dengan cara mengaktivasikan neuron tujuannya.



Gambar 1.1.1 Jaringan Saraf Manusia

Jaringan saraf pada makhluk hidup memiliki jumlah yang sangat banyak sehingga dalam memodelkannya sangat sulit tetapi kita dapat mengadopsi cara kerja dari jaringan saraf dari saat menerima input yang berupa impuls sampai menghasilkan output yang berupa respon. Secara umum, jaringan saraf memiliki input yang berupa impuls, setiap impuls memiliki tingkat kekuatan yang berbeda terhadap suatu neuron atau juga suatu neuron memiliki tingkat respon yang berbeda terhadap input yang berbeda, yang dikenal dengan istilah bobot pada ANN. Kemudian input ini diproses di dalam neuron baru ditransmisikan lagi ke neuron yang lain, proses ini dimodelkan sebagai fungsi aktivasi pada ANN. Yang terakhir adalah output berupa respon yang merupakan hasil dari proses yang sedemikian rupa dan dimodelkan dalam bentuk fungsi matematika di dalam ANN.



Gambar 1.1.2 Struktur Sederhana ANN

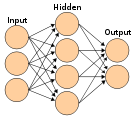
Dalam gambar 1.2 dapat dilihat bahwa terdapat bobot untuk setiap input terhadap salah satu neuron. Untuk suatu objek yang belum pernah disentuh oleh tubuh atau pada ANN proses awal memasukkan input, nilai bobot tersebut akan bersifat random. Dengan mengatur nilai bobot pada ANN , kita dapat membuat sistem tersebut menghasilkan output sesuai dengan yang kita inginkan. Karena jaringan saraf memiliki jumlah neuron yang sangat banyak, begitu juga dengan implementasinya pada ANN, maka pengaturan bobot tersebut cukup sulit dilakukan secara manual. Untuk itu, digunakanlah algoritma yang dapat mengatur bobot pada ANN untuk menghasilkan output yang kita inginkan. Proses pengaturan bobot ini dinamakan proses learning atau training.

Fungsi dari penggunaan ANN antara lain :

* Pattern recognition
* Forecasting
* Data compression
* Dan lain-lain

## **1.2 ARSITEKTUR NEURAL NETWORK**

Pada jaringan ANN, setiap neuron memiliki beberapa masukkan dan keluaran. Input pada ANN dapat berupa data mentah atau juga data hasil olahan dari neuron sebelumnya. Sementara hasil keluaran merupakan hasil akhir dari proses neuron.



Gambar 1.2.1 Skema Neural Network

Pada skema neural network, dapat dilihat bahwa terdapat 3 lapisan utama pada sistem ANN yaitu :

* **Input Layer**

Input layer berfungsi sebagai portal yang menghubungkan jaringan ke sumber data. Neuron yang berada pada input layer tidak dapat memproses data masukan tetapi hanya dapat meneruskan ke neuron pada layer selanjutnya.

* **Hidden Layer**

Hidden layer dapat terdiri dari beberapa layer tergantung dari sistem yang ingin dianalisa. Hidden layer ini berfungsi untuk menerima masukan dari input layer. Masing-masing neuron pada hidden layer ini dipengaruhi oleh semua neuron pada input layer dengan nilai bobot yang berbeda-beda. Pada ANN, setiap neuron pada hidden layer mendapatkan input dari hasil penjumlahan dari perkalian neuron pada input layer dengan bobotnya masing-masing. Setelah neuron pada hidden layer mendapatkan input, maka data tersebut tidak langsung diteruskan ke dalam layer berikutnya tetapi perlu dilakukan suatu proses yang dinamakan proses aktivasi. Proses aktivasi pada ANN ini menggunakan fungsi sigmoid baik itu unipolar maupun bipolar tergantung dari sistem yang ingin dianalisa.

* **Output Layer**

Output layer merupakan layer terakhir pada ANN yang berfungsi untuk menerima input dari layer sebelumnya. Nilai input yang diterima tidak langsung menjadi nilai keluaran sistem tetapi harus melalui proses aktivasi dengan menggunakan fungsi sigmoid.

## **1.3 METODE PEMBELAJARAN**

Pada saat ini, terdapat 3 metode pembelajaran yaitu :

* **Supervised learning**

Pada proses pembelajaran tipe ini, kita diberikan sepasang kumpulan data yaitu input dan target. Pada prosesnya, diinginkan input yang ada dapat dipetakan menjadi target yang ditentukan dengan cara mengatur bobotnya. Pengaturan bobotnya ditentukan oleh cost function yang berhubungan dengan tingkat ketidaksesuaian hasil pemetaan dengan target yang sudah ditentukan. Cost function yang biasa digunakan adalah mean-squarred error. Metode pembelajaran ini biasanya digunakan untuk pattern recognition dan regression.

* **Unsupervised learning**

Pada proses pembelajaran tipe ini, kita diberikan data input dan output yang merupakan suatu fungsi input. Cost function yang digunakan adalah fungsi kuadrat dari selisih input dengan nilai fungsi input. Metode pembelajaran ini biasa digunakan untuk permasalahan estimasi. Contohnya adalah clustering, filtering, compression dan statistical distribution.

* **Reinforcement learning**

Pada proses pembelajaran tipe ini, kita tidak diberikan data input tetapi data tersebut didapatkan dari hasil interaksi langsung dengan lingkungan luar. Pada setiap waktu, secara otomatis akan dihasilkan output, data observasi dan cost function. Biasanya lingkungan dimodelkan dengan Markov Decision Process yang menggunakan probabilitas.

Salah satu algoritma pembelajaran yang sering digunakan adalah algoritma backpropagation yang termasuk ke dalam kategori supervised learning.

## **1.4 IDENTIFIKASI SISTEM MENGGUNAKAN NEURAL NETWORK**

Identifikasi merupakan pemodelan dari sistem kendali yang sudah ada. Identifikasi ini merupakan representasi suatu sistem dengan mencari hubungan dari input dan output dari suatu sistem.

Gambar 1.4.1 Arsitektur Neural Network untuk Sistem Identifikasi

# **BAB 2. BACKPROPAGATION**

## **2.1 ALGORITMA BACKPROPAGATION**

Backpropagation merupakan algoritma pembelajaran untuk memperkecil tingkat error dengan menyesuaikan bobot berdasarkan perbedaan output dan target yang diinginkan.

Proses pembelajaran :

* Pengambilan input
* Indentifikasi error
* Penyesuaian bobot

Algoritma backpropagation :

Step 0 : inisialisasi bobot

* Random

Melakukan inisialisasi bobot secara acak tetapi dalam range angka tertentu. (-x sampai +x)

* Nguyen-Widrow

Metode Nguyen-Widrow sebenarnya juga memakai cara acak tetapi terdapat proses di dalamnya sehingga proses pembelajaran dapat menjadi lebih cepat.

Metode inisialisasi Nguyen-Widrow :

Menentukan faktor skala :

n : jumlah unit input layer

p : jumlah unit hidden layer

β : faktor skala

Algoritma perhitungannya :

1. Inisialisasi bobot secara random

Vij = bilangan acak antara -0.5 sampai 0.5

1. Hitung norm dari setiap bobot pada masing-masing hidden layer
2. Update bobot Vij =
3. Set bias V0j : bilangan acak antara sampai
4. Lakukan langkah 1-4 untuk setiap bobot pada masing-masing output layer yaitu W.

Step 1 : selama kondisi stopping FALSE, lakukan step 2-9

Step 2 : Untuk setiap pasangan pembelajaran, lakukan step 3-8

Step 3 : untuk setiap unit input

* Menerima input Xi
* Mengirimkannya ke semua unit hidden layer

Step 4 : untuk setiap unit hidden layer

* Menghitung semua sinyal input dengan bobotnya :

Z\_inj

* Menghitung nilai aktivasi setiap unit hidden sebagai output unit hidden layer :

Zj = f(z\_inj)

* Mengirimkan nilai aktivasi sebagai input untuk unit output layer

Step 5 : untuk setiap unit output layer

* Menghitung semua sinyal input dengan bobotnya :

y\_ink

* Menghitung nilai aktivasi setiap unit output sebagai output jaringan

Yk = f(y\_ink)

Backpropagation of error

Step 6 : untuk setiap unit output

* Menerima pola target yang bersesuaian dengan pola input
* Menghitung informasi error :

ᵟk = (tk – yk) f’(y\_ink)

* Menghitung besarnya koreksi bobot unit output :

α ∂ E(wjk)/ ∂wjk = α ᵟk zj

* Menghitung besarnya koreksi bias output :

α ᵟk

* Mengirimkan ᵟk ke unit-unit yang ada pada hidden layer

Step 7 : untuk setiap unit hidden

* Menghitung semua koreksi error :
* Menghitung nilai aktivasi koreksi error :

ᵟk = . f’(z\_inj)

* Menghitung koreksi bobot unit hidden :

α ᵟj xi

* Menghitung koreksi error bias unit hidden :

α ᵟj

Step 8 : Setiap unit output

* Update bobot dan biasnya :

wjk(baru) = wjk(lama) + ∆wjk

w0k(baru) = w0k(lama) + ∆w0k

* Setiap unit hidden

vjk(baru) = vjk(lama) + ∆vjk

v0k(baru) = v0k(lama) + ∆v0k

Step 9 : Kondisi stopping ( Toleransi error )

\*Keterangan :

- Toleransi error biasanya bernilai 0.0001

- tk adalah target output yang diinginkan

- v0k dan w0k adalah bobot untuk bias pada hidden layer dan output layer

Pada identifikasi sistem ini digunakan fungsi aktivasi sigmoid bipolar dan normalisasi pada data. Berikut rumus sigmoid bipolar beserta turunannya :

* Fungsi aktivasi sigmoid bipolar
* Turunan fungsi aktivasi sigmoid bipolar

## **2.2 FAKTOR-FAKTOR DALAM PEMBELAJARAN**

Beberapa faktor yang mempengaruhi keberhasilan algoritma backpropagation :

* Inisialisasi bobot

Bobot awal menentukan apakah jaringan akan mencapai global minimum atau local minimum kesalahan dan seberapa cepat jaringan akan konvergen. Inisialisasi secara acak merupakan cara yang paling sering digunakan. Untuk mempercepat proses pelatihan, suatu modifikasi dari inisialisasi acak diperkenalkan oleh Nguyen dan Widrow.

* Laju pembelajaran

Merupakan parameter jaringan dalam mengendalikan proses penyesuaian bobot. Nilai laju pembelajaran yang optimal bergantung pada kasus yang dihadapi. Laju pembelajaran yang terlalu kecil menyebabkan konvergensi jaringan menjadi lebih lambat, sedangkan laju pembelajaran yang terlalu besar dapat menyebabkan ketidakstabilan pada jaringan. Laju pembelajaran memiliki range ideal yaitu 0.2 sampai 0.4.

* Momentum

Momentum digunakan untuk mempercepat pelatihan jaringan. Metode momentum melibatkan penyesuaian bobot ditambah dengan faktor tertentu dari penyesuaian sebelumnya. Penyesuaian ini dinyatakan sebagai :

Momentum ini membuat laju dari perubahan bobot berada pada jalur yang telah dibuat pada saat inisialisasi bobot dan membuat bobot selanjutnya akan berada pada arah jalur yang telah dibuat sebelumnya. Dalam hal ini, momentum digunakan untuk mempercepat proses pembelajaran. Momentum memiliki range dari 0 – 0.9 .

* Local Minimum

Dalam menentukan tingkat error yang paling kecil, backpropagation menggunakan pendekatan secara gradien. Jaringan neural akan mengikuti contour dari error yang dibuat selama proses pembelajaran kemudian mencari titik minimum dari contour tersebut. Untuk jaringan neural yang sederhana misalkan berdimensi dua. Maka untuk mencari nilai minimum cenderung lebih murah karena dinamika contour dari error yang cenderung stabil. Namun untuk jaringan neural yang lebih kompleks, maka dimensi yang dibuat oleh error bukan lagi berdimensi dua melainkan bisa lebih. Dalam jaringan neural yang kompleks, terdapat dinamika error yang dapat menyebabkan kesalahan dalam menentukan titik minimum untuk menandakan proses pembelajaran telah selesai karena titik minimum yang ada bisa merupakan titik local minimum yang dapat terbaca oleh program. Jika program membaca local minimum sebagai titik minimum yang sesungguhnya, maka proses pembelajaran akan seperti berhenti di titik tersebut tetapi sebenarnya titik yang dimaksud tersebut belum mencapai titik minimum yang sebenarnya yaitu global minimum. Kesalahan seperti ini yang seringkali membuat proses pembelajaran menjadi sangat lama dan harus sering diulang.

# **BAB 3. PERCOBAAN**

Pada percobaan kali ini, digunakan algoritma backpropagation dengan ketentuan sebagai berikut :

* + - Input layer memiliki 7 unit neuron dan 1000 variasi data
    - Output layer memiliki 1 unit neuron

## **3.1 VARIASI PERCOBAAN**

Dalam percobaan ini divariasikan banyaknya data input sebanyak 500 sebagai data pembelajaran (*training*) dan 500 data sebagai data percobaan (*testing*).

## **3.2 PARAMETER KEBERHASILAN PERCOBAAN**

Tingkat keberhasilan percobaan ditentukan oleh keluaran (*output*) dari percobaan, apabila keluaran dari percobaan bernilai sama atau mendekati target yang telah ditentukan atau dalam hal ini merupakan keluaran dari sistem maka percobaan dikatakan berhasil.

## **3.3 PROSEDUR PERCOBAAN**

Prosedur percobaan yang saya lakukan adalah

* Untuk semua data pembelajaran dilakukan backpropagation hingga mencapai nilai error yang ditentukan.
* Untuk semua data percobaan dilakukan feedforward dengan mengggunakan bobot yang didapat dari hasil pembelajaran.

## **3.4 PROSES PENGOLAHAN DATA INPUT**

Data yang akan digunakan pada proses pembelajaran merupakan data yang didapat dari plant dan masih berupa data input dan output. Data input telah memiliki persebaran yang merata pada range nilai -1 hingga +1, sedangkan data output masih memiliki distribusi yang tidak merata sehingga sulit sekali untuk diolah. Untuk itu, dilakukan penyetaraan distribusi pada data tersebut dengan menggunakan rumus normalisasi sebagai berikut :

Proses tersebut mengelompokkan data sehingga setiap data nantinya hanya akan berada pada range nilai -1 hingga +1. Setelah proses normalisasi pada output selesai, selanjutnya data disusun agar sesuai dengan ketentuan agar data tersebut dapat digunaakan pada proses pembelajaran. Adapun urutan datanya yakni sebagai berikut

[y(t-1) y(t-2) y(t-3) x(t) x(t-1) x(t-2) x(t-3)]. Setelah semua data diurutkan, barulah kita dapat memasukkan data tersebut ke dalam algoritma backpropagation.

## **3.5 METODE PERCOBAAN**

Dalam melakukan percobaan ini, digunakan software MATLAB. Pada software MATLAB, saya membuat fungsi yang berisi algoritma dari backpropagation di dalam sebuah m file dengan nama backpropagation. Di dalam fungsi tersebut, saya membuat fungsi dengan parameter input, target, alpha, momentum dan hidden layer. Fungsi tersebut mengeluarkan output berupa bobot v,w,v0,w0,jumlah epoch dan nilai error. Selain fungsi tersebut, saya juga membuat fungsi untuk melakukan testing pada data selain dari pembelajaran. Fungsi yang saya buat bernama feedforward dengan parameter v,w,v0 dan w0 yang sudah dihasilkan dari fungsi sebelumnya serta parameter data sisa. Fungsi ini menghasilkan output berupa data yang nantinya akan dibandingkan dengan target yang telah ditentukan.

# **BAB 4. HASIL PERCOBAAN**



Gambar 4.1 Grafik MSSE pada tiap epoch dengan α=0.5, momentum=0.5 dan hidden layer=15



Gambar 4.2 Grafik perbandingan output hasil training dengan target



Gambar 4.3 Grafik perbandingan output hasil testing dengan target

# **BAB 5. ANALISA PERCOBAAN**

Pada saat melakukan proses pembelajaran, hasil yang diinginkan tidak dapat diraih dengan hanya sekali coba, melainkan harus melalui beberapa kali percobaan untuk mendapatkan hasil yang diinginkan. Pada saat melakukan percobaan, terkadang nilai error dari proses pembelajaran mengalami tingkat penurunan yang sangat lambat sehingga proses pembelajaran menjadi sangat lama dan bahkan tidak selesai. Hal tersebut terjadi karena faktor inisialisasi bobot yang telah dijelaskan pada bab 2.2 sebelumnya bahwa inisialisasi bobot mempengaruhi kecepatan pembelajaran jaringan. Setelah dilakukan beberapa kali percobaan, barulah didapatkan data yang diinginkan.

## **5.1 ANALISA GRAFIK**

Pada grafik yang didapatkan, dari grafik pertama dipatkan MSSE sebesar 9.4979x10-6 dan juga dapat dilihat bahwa pada saat awal pembelajaran penurunan MSSE begitu cepat namun ketika pertengahan hingga akhir penurunan tersebut menjadi semakin lambat. Dari grafik kedua terlihat bahwa dari input data training (500 data awal) didapatkan output dari sistem identifikasi bernilai sama dengan output dari plant, hanya saja terdapat sedikit perbedaan di beberapa titik. Dari grafik ketiga dapat dilihat bahwa dari input data testing (500 data terakhir) didapatkan output dari sistem identifikasi juga memiliki nilai yang sama dengan output dari plant dengan MSSE sebesar 9.2347x10-6. Dari kedua grafik tersebut dapat disimpulkan bahwa tujuan dari pembelajaran tercapai, hal tersebut dapat dilihat dari kedua MSSE jika dibandinngkan MSSE pada saat testing lebih kecil dari MSSE pada saat training.

# **BAB 6. KESIMPULAN**

Dari percobaan yang telah dilakukan dan juga hasil percobaan yang didapatkan, dapat disimpulkan beberapa poin yaitu :

* Inisialisasi bobot mempengaruhi kecepatan proses pembelajaran.
* Nilai α mempengaruhi kecepatan proses pembelajaran. Semakin besar α, semakin cepat pula proses pembelajaran. Namun nilai α memiliki range yang membuat proses pembelajaran menjadi ideal. Nilai α yang terlalu besar, menyebabkan terjadinya ketidakstabilan dalam mencapai titik minimum yang berakibat terjadinya osilasi.
* Pengaruh momentum untuk proses update bobot adalah dapat mempercepat proses pembelajaran. Namun hal ini juga tergantung dari proses inisialisasi bobot. Jika proses inisialisasi bobot sudah menghasilkan arah menuju titik minimum yang salah, maka ditambah dengan momentum dapat menghambat proses pembelajaran sehingga memungkinkan terjebak di dalam titik local minimum.
* Untuk setiap nilai α yang sama, proses pembelajaran dapat berbeda karena proses inisialisasi yang bersifat random sehingga setiap kali pembelajaran tidak akan mendapatkan hasil yang sama.
* Model yang didapatkan dari sistem identifikasi tidak akan pernah identik dengan sistem yang sebenarnya, namun perbedaannya dapat diminimalisir dengan membuat MSSE menjadi sekecil mungkin.

# **REFERENSI**

Kusumoputro B., Priandana K., Wahab W. 2015. System Identification and Control of Pressure Process Rig® System Using Backpropagation Neural Networks. ARPN Journal of Engineering and Applied Sciences, Vol. 10, pp. 7190-7195

<http://en.wikipedia.org/wiki/Artificial_neural_network>

<http://catatan-syam.blogspot.com/2013/05/jaringan-syaraf-tiruan-artificial.html>

<http://arxiv.org/ftp/cs/papers/0308/0308031.pdf>

<http://ariesalva.blogspot.com/2009/10/artificial-neural-network-ann.html>

<http://staff.itee.uq.edu.au/janetw/cmc/chapters/BackProp/index2.html>