Paper Jurnal/Prosiding

by Trismayanti Dwi Puspitasari

Penerapan Metode Backpropagation pada Pengenalan Objek Menggunakan Multiple Hidden Layer

Rina Septiriana[#1], Vittalis Ayu[*2], Trismayanti Dwi Puspitasari[#3]

[#]*Jurusan Teknik Informatika dan Komputer Politeknik Negeri Jakarta*
dan Jurusan Teknologi Informasi Politeknik Negeri Jember
Jalan Prof. Dr. G.A. Siwabessy Kampus Baru UI Depok dan Jalan Mastrip POBOX 164 Jember
[1]rinaseptiriana23@gmail.com
[3]trismayantidwipuspitasari@gmail.com

[*] *Program Studi Teknik InformatikaUniversitas Sanata Dharma*
Paingan Maguwoharjo Yogyakarta
[2]vittalisayu@gmail.com

Abstract

Jaringan syaraf tiruan terinspirasi dari cara kerja otak manusia dimana setiap sel syaraf (neuron) memiliki inti sel yang bertugas untuk melakukan pemrosesan informasi. Untuk menentukan jumlah lapisan pada jaringan syaraf merupakan tantangan tersendiri dalam masalah yang kompleks. Namun selain itu penerapan penggunaan multiple hidden layer membutuhkan pemahaman lebih pada algoritma jaringan syaraf tiruan. Pada penelitian ini penulis menggambarkan contoh penerapan multiple hidden layer dalam pengenalan pola objek. Hasil pengujian tersebut menghasilkan nilai akurasi sebesar 50 % dimana dari 6 data yang diuji tersebut hanya 3 buah data yang benar dan sesuai dengan target yang diharapkan.

Keywords— **Backpropagation, Fungsi Aktivasi Sigmoid, Jaringan Syaraf Tiruan, Komputasi Cerdas, Multiple Hidden Layer**

I. PENDAHULUAN

Jaringan syaraf tiruan terinspirasi dari cara kerja otak manusia dimana setiap sel syaraf (neuron) memiliki inti sel yang bertugas untuk melakukan pemrosesan informasi, neuron dikelilingi oleh dendrit yang bertugas menerima informasi dan akson yang merupakan keluaran dari hasil pemrosesan informasi, hasil keluaran dari pemrosesan informasi tersebut dikirimkan antar neuron melalui dendrit [1][2]. Informasi yang diterima oleh neuron lain jika memenuhi batas tertentu yaitu nilai ambang (threshold) dan pada kasus jaringan syaraf tiruan ini dikatakan sebagai kondisi teraktivasi [1].

Beberapa kelas aplikasi yang dapat menerapkan metode jaringan syaraf tiruan antara lain classification, pattern matching, pattern completion, optimization, control, function approximation dan data mining [2]. Dalam penerapannya secara umum jaringan syaraf terbagi mengandung 3 lapisan yaitu lapisan input, lapisan tersembunyi dan lapisan output. Setiap node pada lapisan tersebut mengirimkan sinyalnya dan diaktivasi untuk menghasilkan keluaran dengan nilai error terkecil. Pada penentuan jumlah lapisan tersembunyi tergantung kepada tingkat kerumitan masalah yang dijadikan sebagai studi kasus penelitian. Namun pada dasarnya pengembangan jumlah hidden layer tidak dapat diterapkan pada domain yang terpisah secara linier namun pada fungsi yang mengandung pemetaan berkelanjutan antar finite space [3]. Untuk menentukan jumlah lapisan pada jaringan syaraf merupakan tantangan tersendiri dalam masalah yang

kompleks. Namun selain itu penerapan penggunaan multiple hidden layer membutuhkan pemahaman lebih pada algoritma jaringan syaraf tiruan. Pada penelitian ini penulis menggambarkan contoh penerapan multiple hidden layer dalam pengenalan pola objek bangun persegi dengan menggunakan 2 unit input, 2 buah lapisan tersembunyi dengan 3 buah node di setiap lapisan, 1 lapisan input, Learning rate (α), 20 buah data training.dan menggunakan fungsi aktivasi sigmoid.

II. LANDASAN TEORI

Jaringan syaraf tiruan merupakan representasi buatan dari otak manusia yang mencoba untuk mensimulasikan proses pembelajaran pada otak manusia, istilah tiruan dimaksudkan untuk menggambarkan bahwa jaringan syaraf tiruan diimplementasikan dalam program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran. [1]

A. Arsitektur Jaringan Syaraf Tiruan

Arsitektur pada jaringan syaraf tiruan terdiri dari:[4]

a. Jaringan dengan lapisan syaraf tunggal (single layer net). Jaringan ini hanya menerima input kemudian secara langsung akan diolah menjadi output tanpa melewati lapisan tersembunyi.
b. Jaringan dengan banyak lapisan (multiple net) memiliki satu atau lebih lapisan yang terletak diantara lapisan input dan lapisan output yatu 1 atau lebih lapisan tersembunyi. Pembelajaran pada jaringan ini lebih sukses dalam menyelesaikan permasalahan.

B. Fungsi Aktivasi

Pada jaringan syaraf tiruan terdapat beberapa fungsi aktifasi yang digunakan antara lain:

a. Fungsi Undak Biner, pada fungsi ini output biner yang dihasilkan 0 dan 1.
b. Fungsi Bipolar, pada fungsi ini output yang dihasilkan berupa 1, 0 atau-1.
c. Fungsi Linier, pada fungsi ini nilai output yang dihasilkan sama dengan nilai outputnya.
d. Fungsi sigmoid biner, nilai output yang dihasilkan terletak dalam rentang nilai 0 dan 1, yang dapat dirumuskan sebagai berikut:

$$y = f(x) = \frac{1}{1 + e^{-\sigma x}}$$

III. METODE PENELITIAN

Metode penelitian yang digunakan di sini adalah dengan menggunakan metode backpropagation. Langkah – langkah yang dilakukan adalah sebagai berikut: [4]

- Inisialisasi bobot dan bias dengan merandom bilangan antara 0 dan 1, untuk tiap pasangan elemen yang akan dilakukan pembelajaran, lakukan:
- Feedforward

a. Setiap unit input (Xi, i=1,2,3,..,n) menerima sinyal xi dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).
b. Setiap unit lapisan tersembunyi pertama (Yai, i=1,2,3,…,n) menjumlahkan sinyal-sinyal input terbobot:

$$ya\_in_j = v0_j + \sum_{i=1}^{n} x_i v_{ij}$$

dan kemudian gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$$ya_j = f(ya\_in_j)$$

Setelah itu hasilnya digunakan untuk mengirim sinyal output ini ke seluruh unit pada unit hidden layer yang kedua.

c. Setiap unit lapisan tersembunyi kedua (Ybi, i=1,2,3,…,n) menjumlahkan sinyal-sinyal hidden layer pertama yang terbobot:

$$yb\_in_j = u0_j + \sum_{i=1}^{n} ya_i u_{ij}$$

dan kemudian gunakan fungsi aktivasi untuk menghitung sinyal outputnya pada lapisan tersembunyi berikutnya:

$$yb_j = f(yb\_in_j)$$

Setelah itu hasilnya digunakan untuk mengirim sinyal output ini ke seluruh unit pada unit output.

d. Setiap unit output (zj, j=1,2,3,…,p) menjumlahkan sinyal-sinyal input terbobot:

$$z\_in_k = w0_k + \sum_{j=1}^{n} yb_j w_{jk}$$

dan kemudian gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$$z_j = f(z\_in_j)$$

lalu hasilnya digunakan untuk mengirim sinyal output ini ke seluruh unit pada unit output.

- Backpropagation

e. Setiap unit output (yk, i=1,2,3,…,m) menerima target pola yang berhubungan dengan pola input pembelajaran, hitung informasi errornya:

$$\delta_k = (t_k - y_k) f'(z\_in_k)$$

Kemudian hitung koreksi bobot yang nantinya akan digunakan untuk memperbaiki nilai wjk:

$\Delta wjk = \alpha \delta kzj$

Hitung juga koreksi bias yang nantinya akan digunakan untuk memperbaiki nilai w0k:

$\Delta w0k = \alpha \delta k$

Faktor δk ini kemudian dikirimkan ke layer yang berada di bawahnya.

f. Setiap unit lapisan tersembunyi kedua (zk i=1,2,3,…,m) menjumlahkan delta inputnya dari unit-unit yang berada pada lapisan di atasnya:

$$\delta\_in_j = \sum_{k=1}^{m} \delta_k w_{jk}$$

.Kemudian kalikan nilai tersebut dengan turunan dari fungsi aktivasinya untuk menghitung informasi yang error:

δj= δ_inj f'(yb_inj)

Hitung koreksi bobot yang nantinya akan digunakan untuk memperbaiki nilai vij:

Δuij=αδjyai

hitung juga koreksi bias yang nantinya akan digunakan untuk memperbaiki nilai v0j:

Δu0j=αδj

g. Setiap unit lapisan tersembunyi pertama (zk i=1,2,3,…,m) menjumlahkan delta inputnya dari unit-unit yang berada pada lapisan di atasnya:

$$\delta\_in_j = \sum_{k=1}^{m} \delta_k u_{jk}$$

Kemudian kalikan nilai tersebut dengan turunan dari fungsi aktivasinya untuk menghitung informasi yang error:

δj= δ_inj f'(ya_inj)

Hitung koreksi bobot yang nantinya akan digunakan untuk memperbaiki nilai vij:

Δvij=αδjxi

hitung juga koreksi bias yang nantinya akan digunakan untuk memperbaiki nilai v0j:

Δv0j=αδj

h. Setiap unit output (yk, k=1,2,3,…,n) memperbaiki bias dan bobotnya (k=1,2,3,…,n):

wjk(baru)=wjk(lama) + Δwjk

Demikian pula untuk setiap hidden unit untuk lapisan pertama akan memperbaharui bias dan bobotnya dari setiap unit input.

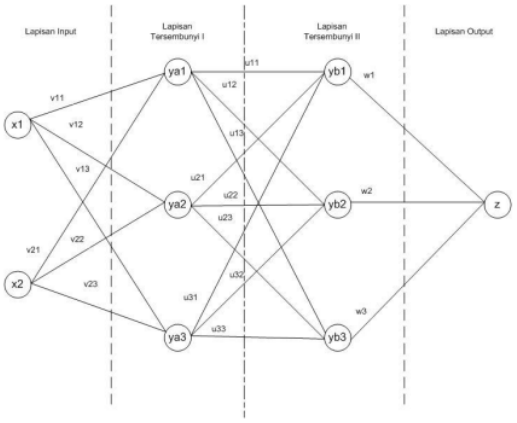
vij(baru)=vij(lama) + Δvij

Demikian pula untuk setiap hidden unit untuk lapisan kedua akan memperbaharui bias dan bobotnya dari setiap unit lapisan tersembunyi pertama.

uij(baru)=uij(lama) + Δuij

IV. HASIL DAN PEMBAHASAN

Untuk menyelesaikan permasalahan berdasarkan pada rumusan masalah yang telah dipaparkan sebelumnya, maka langkah pertama yang akan dilakukan adalah merancang arsitektur yang akan dipergunakan di dalam program yang akan dibangun. Adapun rancangan arsitekturnya dapat dilihat pada Gambar 1.



Gambar 1 Arsitektur Jaringan Syaraf Tiruan Pengenalan Persegi

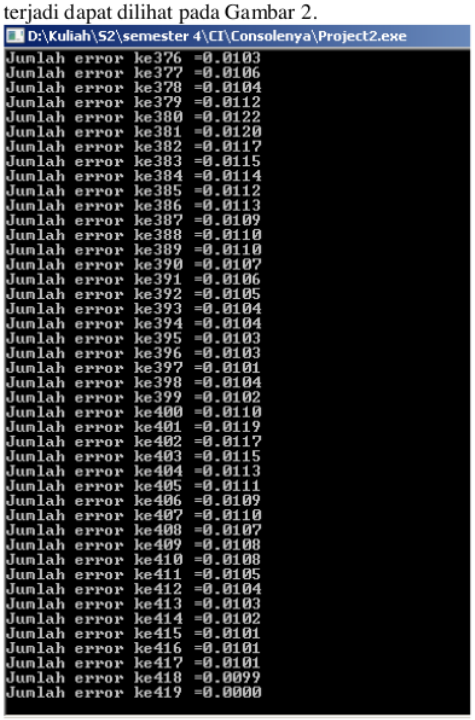
Dari arsitektur tersebut terbentuklah sebuah algoritma yang sudah dijabarkan sebelumnya pada metode penyelesaian. Dimana setiap unit yang ada merupakan hasil penambahan dari unit yang ada sebelumnya. Selain itu di dalam arsitektur tersebut juga ditambahkan unit bias untuk membantu perhitungan. Unit bias tersebut antara lain adalah v0i (unit bias di antara lapisan input dan lapisan tersembunyi), u0j (unit bias di antara lapisan tersembunyi pertama dan kedua), dan w0 (unit bias di antara lapisan tersebunyi kedua dan lapisan output).

Setelah dirancang arsitekturnya kemudian dilakukan proses pengkodingan dimana setiap tahapan proses diterjemahkan ke dalam Delphi Console (code terlampir). Adapun data training yang akan dimasukkan ke dalam program dapat dilihat pada Tabel 1.

TABEL 1
DATA TRAINING

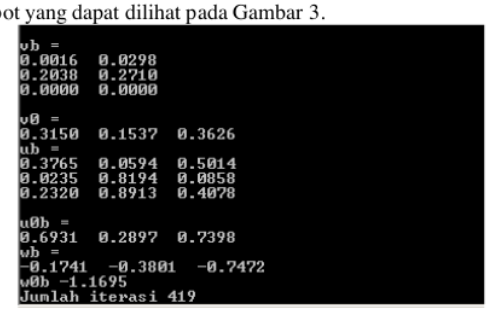
No.	x1	x2	Target (t)
1.	0.1	0.1	1
2.	0.2	0.2	1
3.	0.3	0.3	1
4.	0.4	0.4	1
5.	0.5	0.5	1
6.	0.6	0.6	1
7.	0.7	0.7	1
8.	0.8	0.8	1
9.	0.45	0.45	1
10.	0.9	0.9	1
11.	0.1	0.9	0
12.	0.2	0.45	0
13.	0.3	0.8	0
14.	0.4	0.7	0
15.	0.5	0.6	0
16.	0.6	0.5	0
17.	0.7	0.4	0
18.	0.8	0.3	0
19.	0.45	0.2	0
20.	0.9	0.1	0

Data training tersebut kemudian dilatih dengan menggunakan program yang telah dibuat. Perubahan error yang terjadi dapat dilihat pada Gambar 2.



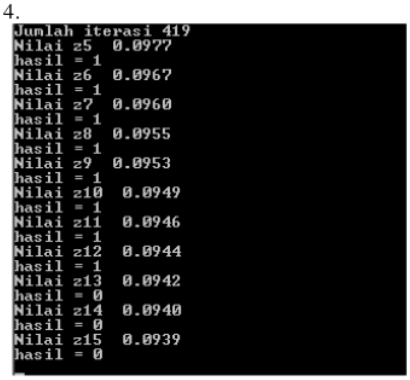
Gambar 2 Perubahan Error di program

Dari Gambar 2 terlihat penurunan jumlah error dari iterasi ke-376 sampai dengan iterasi ke -419 (iterasi terakhir) dengan rentang perubahan error mulai dari 0.0103 sampai dengan 0.0000 (pembulatan 4 angka di belakang koma). Dengan proses training tersebut didapatkan perubahan nilai bobot yang dapat dilihat pada Gambar 3.



Gambar 3 Nilai Bobot Terakhir

Nilai Bobot terakhir didapatkan setelah melakukan proses training data sampai dengan iterasi ke 419, dan nilai bobot pada Gambar 3 tersebut juga digunakan untuk menentukan data threshold dari data training pada Tabel 1 dan didapatkan nilai threshold sebesar 0.0942. Setelah didapatkan nilai thresholdnya kemudian dilakukan pengujian dengan menggunakan data training ke-5 sampai ke-15 (10 buah data training). Hasilnya dapat dilihat Gambar 4.

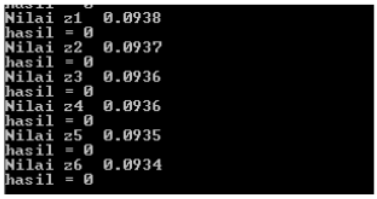


Gambar 4 Hasil Pengujian dari 10 data training

Dari pengujian tersebut jika dibandingkan dengan Tabel 1, terdapat 2 buah data yang salah dimana seharusnya data ke-11 dan data ke-12 bernilai 0 (bukan termasuk persegi). Kemudian program tersebut diuji dengan menggunakan data uji yang dapat dilihat pada Tabel 2 dan hasil pengujian dari data uji tersebut dapat dilihat pada Gambar 5.

TABEL 2
DATA UJI

No.	x1	x2	Target (t)
1.	0.4	0.3	0
2.	0.2	0.8	0
3.	0.3	0.3	1
4.	0.25	0.25	1
5.	0.6	0.7	0
6.	0.33	0.33	1



Gambar 5 Hasil Pengujian dari Data Uji

Hasil pengujian tersebut menghasilkan nilai akurasi sebesar 50 % dimana dari 6 data yang diuji tersebut hanya 3 buah data yang benar dan sesuai dengan target yang diharapkan.

V. KESIMPULAN.

Berdasarkan percobaan program yang dilakukan didapatkan beberapa kesimpulan sebagai berikut:

a. Jumlah lapisan pada hidden layer mempengaruhi jumlahnya iterasi. Semakin banyak lapisan tersembunyi

yang digunakan maka semakin cepat proses iterasi yang terjadi.

b. Jumlah data training mempengaruhi akurasi dari program yang dibuat, selain itu data training yang tepat juga mempengaruhi akurasi hasil keluaran yang didapat.
c. Penentuan nilai threshold harus diperhatikan sebaik mungkin sehingga hasil yang didapatkan juga semakin akurat.

DAFTAR PUSTAKA

[1] Kusumadewi, Sri. 2003. Artificial Intelligence (Teknik dan Aplikasinya). Yogyakarta: Graha Ilmu.

[2] Engelbrecht, Andries P. 2007. Computational Intelligence an Introduction Second Edition. Great Britain: John Willey and Sons Ltd.

[3] Karsoliya, Saurabh. 2012. "Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture," International Journal of Engineering Trends and Technology- Volume 3 Issue 6- 2012.

[4] S. Zhang, C. Zhu, J. K. O. Sin, &P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," *IEEE Electron Device Lett.*, vol. 20, pp. 569–571, Nov. 1999.

[5] Fausett, Lauren.1994. Fundamentals Of Neural Networks: Architectures, Algorithms,and Applications.., New Jersey, USA: Prenctice-Hall, Inc.

Paper Jurnal/Prosiding

ORIGINALITY REPORT

14%	14%	16%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	informatikaunindra.org Internet Source	3%
2	lib.uin-malang.ac.id Internet Source	3%
3	nanopdf.com Internet Source	3%
4	fr.scribd.com Internet Source	3%
5	vdocuments.site Internet Source	2%

Exclude quotes	On	Exclude matches	< 2%
Exclude bibliography	On		