

ESTIMASI JUMLAH PASIEN GAWAT DARURAT DENGAN METODE PROPAGASI BALIK PADA R.S ISLAM FAISAL MAKASSAR

Muhammad Faisal

Program Studi Manajemen Informatika, STMIK Profesional Makassar

muh.faisal.art@gmail.com

Abstrak

Rumah Sakit pada bagian Unit Gawat Darurat (UGD) memiliki peran penting dalam proses penanganan pasien gawat darurat. Ketersediaan fasilitas untuk dapat mendukung tenaga medis dalam bekerja merupakan faktor yang sangat signifikan sehingga pelayanan yang lebih optimal dapat terlaksana. Dalam rangka pencapaian tingkat pelayanan yang maksimal dipandang perlu mempertimbangkan kesesuaian antara ketersediaan sumber daya dengan jumlah pasien UGD yang akan ditangani.

Jaringan Saraf Tiruan (JST) Propagasi Balik merupakan salah satu sistem pemrosesan informasi yang didesain dengan menirukan cara kerja otak manusia dalam menyelesaikan suatu masalah melalui proses pembelajaran terhadap setiap neuron, layer dan bobot yang telah ditentukan sehingga mampu membantu pengguna melakukan prediksi jumlah pasien gawat darurat dalam setiap waktu. Ketersediaan data yang objektif akan mempengaruhi akurasi terhadap hasil yang diperoleh. Diharapkan melalui aplikasi yang dibangun dapat membantu pihak rumah sakit dalam mempersiapkan sumber daya sehingga setiap pasien UGD dapat ditangani dengan baik.

Kata kunci : Jaringan Saraf Tiruan, Propagasi Balik, Estimasi Pasien.

A. PENDAHULUAN

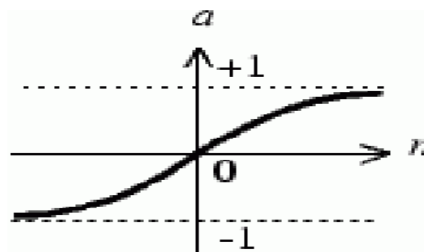
Rumah sakit merupakan salah satu bentuk sarana kesehatan, baik yang diselenggarakan oleh pemerintah dan atau masyarakat yang berfungsi untuk melakukan upaya kesehatan dasar atau kesehatan rujukan dan upaya kesehatan penunjang. Rumah sakit dalam menjalankan fungsinya diharapkan senantiasa memperhatikan fungsi sosial dalam memberikan pelayanan kesehatan pada masyarakat. Keberhasilan rumah sakit dalam menjalankan fungsinya di tandai dengan adanya mutu pelayanan prima rumah sakit. Rumah Sakit khususnya di Unit Gawat Darurat (UGD) memiliki peran sebagai gerbang utama jalan masuknya penderita gawat darurat. Kemampuan suatu fasilitas kesehatan secara keseluruhan dalam kualitas dan kesiapan dalam perannya sebagai pusat rujukan penderita dari pra rumah sakit tercermin dari kemampuan unit gawat darurat.

Masalah yang di hadapi pada unit gawat darurat adalah susahnya untuk memprediksi jumlah pasien minggu berikutnya. Karena permasalahan di atas maka diperlukan sebuah sistem yang membantu agar dapat memprediksi jumlah pasien gawat darurat untuk minggu

selanjutnya melalui beberapa acuan sebagai parameter. Untuk mengolah dan menganalisis parameter tersebut di gunakan metode jaringan saraf tiruan (JST) propagasi balik. Karena JST adalah sistem pemrosesan informasi yang didesain dengan menirukan cara kerja otak manusia dalam menyelesaikan suatu masalah dengan melakukan proses belajar melalui perubahan bobot. JST propagasi balik adalah algoritma pembelajaran yang terawasi (supervised training). Arsitektur jaringan yang digunakan biasanya multi-layer (multi layer perceptron). Algoritma propagasi balik menggunakan error output untuk mengubah nilai bobot-bobotnya dalam arah mundur sehingga dapat menghasilkan solusi yang lebih baik.

B. METODE PENELITIAN

Algoritma Propagasi Balik merupakan algoritma pembelajaran yang terawasi (supervised Training). Arsitektur jaringan yang digunakan biasanya multi-layer (multi layer perceptron), algoritme tersebut menggunakan error output untuk mengubah nilai bobot-bobotnya dalam arah mundur (propagasi balik). Untuk mendapatkan error ini, tahapan yang pertama dilakukan adalah tahap perambatan maju (forward propagation). Fungsi aktivasi yang biasa digunakan adalah fungsi sigmoid.



Gambar 1. Fungsi signoid

Pelatihan propagasi balik meliputi 3 fase sebagai berikut :

- a. Fase 1, yaitu propagasi maju (*feedforward*).
Pola masukan dihitung maju mulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yang ditentukan.
- b. Fase 2, yaitu propagasi mundur (*backward*).
Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan yang terjadi itu dipropagasi mundur. Dimulai dari garis yang berhubungan langsung dengan unit-unit di layar keluaran.
- c. Fase 3, yaitu perubahan bobot.
Modifikasi bobot untuk menurunkan kesalahan yang terjadi. Ketiga fase tersebut

diulang-ulang terus hingga kondisi penghentian dipenuhi.

Algoritma pelatihan untuk jaringan *backpropagation* dengan tiga layer tersembunyi (dengan fungsi aktivasi sigmoid biner) adalah sebagai berikut :

Langkah 0 : Pemberian insiasialisasi bobot untuk masing-masing input (diberi nilai secara acak).

Langkah 1 : Jika kondisi tidak tercapai, lakukan langkah 2-9.

Langkah 2 : Untuk setiap pasangan pelatihan, lakukan langkah 3-8.

Feedforward:

Langkah 3 : Tiap unit masukan ($x_i, i = 1, \dots, n$) menerima sinyal x_i dan menghantarkan sinyal ini ke semua unit lapisan di atasnya (unit tersembunyi).

Langkah 4 : Setiap unit tersembunyi ($x_i, i = 1, \dots, p$) jumlahkan bobot sinyal

masukannya, $z_{inj} = v_{oj} + \sum_{i=1}^n x_i v_{ij}$ v_{oj} = bias pada unit tersembunyi j

aplikasikan fungsi aktivasinya untuk menghitung sinyal keluarannya,

$$z_j = f(z_{inj}) = \frac{1}{1 + e^{-z_{inj}}}, \text{ dan dikirimkan sinyal ini keseluruhan unit pada}$$

lapisan di atasnya (unit keluaran).

Langkah 5 : Tiap unit keluaran ($y_k, k = 1, \dots, m$) mejumlahkan bobot sinyal masukannya,

$$y_{ink} = w_{ok} + \sum_{j=1}^n z_j w_{jk} \quad w_{ok} = \text{bias pada unit keluaran } k \text{ dan}$$

aplikasikan fungsi aktivasinya untuk menghitung sinyal keluarannya,

$$y_k = f(y_{ink}) = \frac{1}{1 + e^{-y_{ink}}}.$$

Backward:

Langkah 6 : Tiap unit keluaran ($y_k, k = 1, \dots, m$) menerima pola target yang saling berhubungan pada masukan pola pelatihan, hitung kesalahan informasinya,

$$\delta_k = (t_k - y_k) f'(y_{ink}) \text{ hitung koreksi bobotnya (digunakan untuk}$$

memperbarui w_{jk} nantinya), $\Delta w_{jk} = \alpha \delta_k z_j$ hitung koreksi biasnya

(digunakan untuk memperbarui w_{ok} nantinya), dan dikirimkan ke unit-unit pada lapisan bawahnya.

Langkah 7 : Setiap unit lapisan tersembunyi $z_j, j = 1, \dots, p$) jumlahkan hasil perubahan

masukannya (dari unit-unit lapisan atasnya), $\delta_{inj} = \sum_{k=1}^m z \delta_k w_{jk}$ kalikan

dengan turunan fungsi aktivasinya untuk menghitung informasi

kesalahannya, $\delta_j = \delta_{in_j} f^1(z_{in_j})$ koreksi bobot v_{ij} , $\Delta v_{ij} = \alpha \delta_j x_i$ dan koreksi bias v_{0j} . $\Delta v_{0j} = \alpha \delta_j$ Lanjutkan ke tahap peng-*update*-an weight.

Langkah 8 : Tiap unit keluaran ($y_k, k = 1, \dots, m$) *update* bias bobotnya ($j = 0, \dots, p$)

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}$$

Tiap unit lapisan tersembunyi ($z_j, j = 1, \dots, p$) *update* bias dan bobotnya ($I =$

$$0, \dots, n) : v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}$$

Langkah 9 : Test kondisi, jika *true* maka training berhenti.

$f(x)$ = fungsi *aktivasi*

$x_{\text{setb}} = (x_1, x_2, \dots, x_n)$

n = jumlah *neuron* pada *input layer*

l = banyaknya pasangan data pelatihan

p = jumlah *neuron* pada *hidden layer*

k = jumlah *neuron* pada *output layer*

v = bobot *input layer* ke *hidden layer*

w = bobot *hidden layer* ke *output layer*

z_{in} = inyal *input hidden neuron*

y_{in} = sinyal *input output neuron*

x = sinyal *input*.

Z = sinyal *output / level aktivasi* dari *hidden neuron*

Y = sinyal *output / level aktivasi* dari *output neuron*

α = parameter *learning rate*

δ_k = faktor kesalahan unit *output k*.

C. HASIL

Adapun hasil perancangan dan pengujian terhadap aplikasi yang digunakan adalah sebagai berikut :

Input Data Training dan Pengujian

Gambar 2. Data Input dan Pengujian

Data Hasil Normalisasi.

Gambar 3. Hasil Normalisasi

Pengujian dan Hasil Iterasi I.



Gambar 4. Pengujian dan hasil iterasi 1

Pengujian dan Hasil Iterasi II



Gambar 5. Pengujian dan hasil iterasi 2

Pengujian dan Hasil Iterasi III



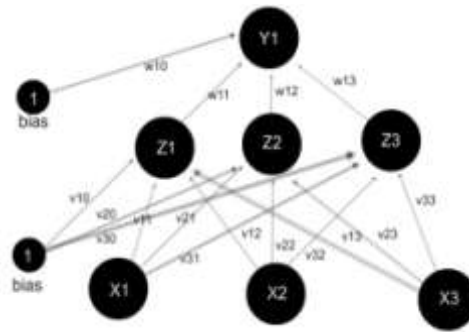
Gambar 6. Pengujian dan hasil iterasi 3

Pengujian dan Hasil Iterasi IV

Gambar 7. Pengujian dan hasil iterasi 5

D. PEMBAHASAN

Sebelum dilakukan penerapan metode pada aplikasi yang dibangun terlebih dahulu membangun Arsitektur program yang dapat dilihat pada gambar 8.



Gambar 8. Arsitektur program

X1= Pasien Gawat darurat pada 2 minggu yang lalu

X2= Pasien Gawat darurat pada seminggu yang lalu

X3= Pasien Gawat darurat pada minggu ini

T = Pasien Gawat darurat yang akan datang.

Cara kerja program:

1. Menginisialisasi bobot untuk setiap inputan
2. Mengitung nilai yang masuk ke layer tersembunyi
3. Melakukan aktivasi terhadap nilai yang masuk ke layer tersembunyi
4. Mengitung nilai yang masuk ke layer output
5. Melakukan aktivasi terhadap nilai yang masuk ke layer output
6. Melakukan pembaharuan bobot dari layer input ke layer tersembunyi.

7. Melakukan pembaharuan bobot dari layer tersembunyi ke layer output.

Program akan menentukan banyaknya pasien gawat darurat untuk minggu yang akan datang berdasarkan jumlah pasien gawat darurat dalam 3 minggu terakhir. Sehingga program JST ini menggunakan 3 variabel input yaitu jumlah pasien gawat darurat 2 minggu yang lalu, pasien gawat darurat seminggu yang lalu dan pasien gawat darurat pada minggu ini. Program ini menggunakan 3 buah hidden layer dan satu buah output layer.

Sebelum dilakukan training, data yang akan ditraining harus dinormalisasi terlebih dahulu menggunakan rumus : $Normalisasi = \frac{Data - DataMin}{DataMax - DataMin}$

Pada gambar 9 ditampilkan data tentang Jumlah pasien UGD setiap hari dalam satu bulan terakhir secara acak.

Input Data Pasien				
	X1	X2	X3	T
Data 1	23	25	22	26
Data 2	23	25	24	21
Data 3	25	21	25	22
Data 4	26	24	21	21
Data 5	24	25	23	24
Data 6	21	23	22	25
Data 7	26	22	24	27

Gambar 9. Input data pasien

Setiap inputan data membentuk pola inputan yang akan digunakan untuk pengujian. Kemudian dari hasil pengujian tersebut akan dikeluarkan sebuah output yaitu jumlah pasien UGD, yang diperkirakan nilainya akan sama dengan jumlah pasien UGD yang akan datang, sehingga pihak rumah sakit dapat melakukan persiapan dan antisipasi mengenai hal yang berhubungan dengan pasien Unit Gawat Darurat. Setelah melalui beberapa tahapan iterasi pada proses algoritma propagasi balik maka akan tampil nilai yang dijadikan sebagai perkiraan jumlah pasien yang akan datang. Hasil akhir dapat dilihat pada gambar 10.

PROSES HASIL		
	Target Awal	Target Hasil
Data 6	23.0	24.0
Data 7	23.0	24.0

Gambar 10. Hasil akhir Propagasi Balik.

E. KESIMPULAN

Dari data dan percobaan diatas dapat disimpulkan bahwa:

1. Program perkiraan jumlah pasien gawat darurat ini menggunakan 3 input layer, 3 hidden layer dan satu output layer.
2. Berdasarkan proses training dan proses pengujian, program ini dapat digunakan untuk menentukan jumlah pasien gawat darurat untuk minggu yang akan datang.
3. Berdasarkan hasil training dan pengujian, tingkat keakuratan program berbanding lurus dengan nilai learning rate yang digunakan.
4. Tingkat keakuratan program juga dipengaruhi oleh nilai bobot awal untuk setiap inputan dan jumlah epoch yang digunakan untuk proses training.

DAFTAR PUSTAKA

- [1] SriKusumadewi. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Graha Ilmu. Yogyakarta.
- [2] <http://www.scribd.com/doc/33641732/Backpropagation-Neural-Network-for-XOR-Problem-Java-source-code> diakses pada 12 Desember 2012.
- [3] <http://xa.yimg.com/kq/groups/20972564/1923605698/name/Neural+Network+6.ppt>. diakses pada 10 Desember 2012.
- [4] <http://lecturer.ukdw.ac.id/anton/download/AI/Jaringan%20Saraf%20Tiruan.ppt>. diakses 10 Desember 2012.
- [5] Wikipedia .2012. *Jaringan saraf tiruan*. http://id.wikipedia.org/wiki/Jaringan_saraf_tiruan, di akses_pada tanggal 10 Desember 2012.