PERBANDINGAN PERFORMANSI METODE *BP* DAN *RBF* UNTUK PERMASALAHAN KLASIFIKASI PENYAKIT DIABETES MELLITUS

Adisty Sukmawati[1]
[1] Teknik Informatika – Universitas Komputer Indonesia
Jl. Dipatiukur 112 – 114 Bandung
E-mail : adistysu@gmail.com[1]

ABSTRAK

Jaringan syaraf tiruan merupakan salah satu metode yang dapat digunakan untuk melakukan klasifikasi suatu data.Metode jaringan syaraf tiruan yang sering digunakan untuk mengklasifikasikan penyakit diantaranya *Backpropagation* dan *Radial Basis Function*. *Backpropagation* menggunakan nilai *error* untuk mengubah bobot – bobotnya dalam dua arah yaitu arah mundur dan maju sehingga mendapat output yang baik. *Radial Basis Function* melakuan pembelajaran dengan dua tahap yaitu terawasi dan tidak terawasi.

Penyakit diabetes merupakan penyakit yang umum yang di derita oleh masyarakat. Penyakit diabetes dibagi menjadi dua jenis yaitu diabetes mellitus insulin dan diabetes mellitus non insulin. Penanganan masing – masing jenis diabetes mellitus ini berbeda – beda, perlu adanya pemeriksaan sejak dini agar setiap pasien diabetes mellitus mendapat penanganan yang sesuai dengan jenis diabetes yang di derita.

Berdasarkan hasil pengujian yang terlah dilakukan metode *radialbasis function* memiliki nilai error yang lebih sedikit dibandingkan metode *backpropagation.*

Kata kunci : *Backpropagation , Radial Basis Function, k-fold cross validation*

1. PENDAHULUAN

Mengidentifikasi ciri suatu objek merupakan salah satu cara untuk mendeskripsikan atau mengklasifikasikan suatu data kelompok tertentu berdasarkan kesamaan karakteristik yang ada dalam suatu objek. Cara melakukan klasifikasi salah satunya dapat menggunakan metode dalam jaringan syaraf tiruan. Metode jaringan syaraf tiruan yang sering digunakan untuk mengklasifikasikan penyakit diantaranya *Backpropagation* dan *Radial Basis Function*. Kedua metode ini memiliki ciri khas masing – masing dalam melakukan klasifikasi. *Backpropagation* menggunakan nilai *error* untuk mengubah bobot – bobotnya dalam dua arah yaitu arah mundur dan maju sehingga mendapat output yang baik. *Radial Basis Function* melakuan pembelajaran dengan dua tahap yaitu terawasi dan tidak terawasi, maksud tidak terawasi terdapat algoritma lain untuk membantu dalam proses pembelajaran dalam menentukan kebutuhan dalam proses pembelajaran dan melakukan algoritma radial basis function untuk proses selanjutnya dalam proses pembelajaran.

Karena metode *Backpropagation* dan *Radial Basis Function* sering digunakan untuk klasifkasi penyakit. Penyakit diabetes mellitus membutuhkan klasifikasi agar dalam melakukan tindak lanjut didapat penanganan yang tepat berdasarkan klasifikasi jenis diabetes mellitus.

1.1 Backpropagation Neural Network

Backpropagation Neural Network merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot – bobot yang terhubung dengan *neuron – neuron* yang ada pada lapisan tersembunyinya. Algoritma *backpropagation neural netwrok* menggunakan *error output* untuk mengubah nilai bobot – bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu.

1.2 Radial Basis Function

Jaringan Radial Basis Function adalah salah satu contoh JST dengan lapis banyak dengan arah perambatan pembelajaran maju (Multilayer FeedForward Neural). Jaringan ini memiliki 3 buah lapisan yaitu lapisan input,tersembunyi dan output. Pada Radial Basis Function , lapisan tersembunyi hanya ada 1 buah saja.

Radial Basis Function ini memiliji nilai maksismum 1, yang terjadi apabila input yang diterima bernilai 0 (jarak antara bobot dengan input 0 . sehingga bila jarak antar bobot dengan input berkuran, fungsi ini akan menghasilkan output yang lebih besar. Umunya fungsi aktivasi Radial Basis Function menggunakan fungsi Gaussian dan linier pada lapisan output. Berikut adlaah fungsi aktivasi Gaussian:

$$\varphi\left(\left|\left|\bar{x}-\bar{c_i}\right|\right|\right)=\exp\left(\frac{\|x-\ c_i\|^2}{2\sigma_j{}^2}\right)$$

Dimana :
x = input

ci = pusat data ke – i , i = 1, 2, ..., n
$\sigma_j{}^2$ = standar deviasi ke – j , j = 1, 2,..., n
n = banyak pusat data

2.4 Ukuran Ketepatan Pengujian

Untuk mengukur suatu pengujian dapat dilakukan dengan menghitung selisih antara data hasil perhitungan dengan data tujuan aktual. Selisih besaran (ukuran kesalahan pengujian) data pengujian terhadap data aktual yang terjadi merupakan suatu data penring untuk menilai ketepatan suatu metode pengujian. Dengan membandingkan ukuran kesalahan beberapa metode pengujian, akan diperoleh metode mana yang mempunyai ukuran kesalahan terkecil, sehingga nilai pengujian dapat dipakai sebagai acuan dalam menentukan kebutuhan – kebutuhan dimana yang akan datang. Terdapat beberapa ukuran kesalahan dalam pengujian antara lain.

- *Root Mean Squared Error* (*RMSE*)
  RMSE merupakan nilai tengah kesalahan akar kuadrat dengan persamaan sebagai berikut :

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_t - (F_t))^2}{n}}$$

RMSE merupakan metode alternatif untuk mengevaluasi teknik peramalan. *RMSE* adalah rata – rata kuadrat dari perbedaan nilai estimasi dengan nilai obeservasi suatu variabel. Jika nilai *RMSE* semakin kecil maka estimasi model atau variabel tersebut semakain valid. Notasi dasar dari persamaan diatas secara ringkas adalah sebagai berikut :
X_t= nilai aktual pada periode t
F_t = nilai pengujian pada periode t
$X_t - F_t$= nilai kesalahan pengujian
n = jumlah data

2. ISI PENELITIAN

Dalam kasus ini kedua metode antara metode *Backpropagation* dan *Radial Basis Function* measing – masing melakukan pembelajaran dan pengujian yang dilakukan untuk tuajuan mengetahui perbandingan performansi mana yang lebih baik.

2.1 Algoritma Metode Backpropagation

Algoritma metode *BackpropagationNeural Network* :
Menetapkan bobot (w), *learning rate* (α), maksimum *epoch* (MaxEpoch).
Menentukan inisialisasi bobot (ambil nilai random yang cukup kecil)

1. Bobot awal input ke *hidden* layer [v_{ij}]
2. Bobot awal bias ke *hidden* layer [v_{0j}]
3. Bobot awal *hidden* layer ke output layer [w_j]
4. Bobot awal bias ke output layer [w_0]

Kerjakan langkah – langkah berikut selama kondisi berhenti benilai *FALSE* :

1. Untuk tiap –tiap *input layer* yang akan dilakukan pembelajaran, kerjakan :

Feedforward :
Langkah 1 : menghitung nilai yang didapat dari sinyal input menuju kehidden layer dengan menggunakan persamaan :

$$z_in_j = v0_j + \sum_{i=1}^{n} x_i\, v_{ij}$$

Dimana :
z_in_j = nilai dari input menuju *hidden layer*
$v0_j$ = bobot awal bias input menuju *hidden layer*
x_i = nilai input layer ;
x1 = tinggi badan
x2 = berat badan
x3 = tensi
x4 =gula darah sebelum puasa (GDP)
x5 = gula darah sesudah puasa (GDPP)
v_{ij} = nilai bobot awal *hidden layer* ke *hidden layer*
Langkah 2 : menghitung fungsi aktivasi pada sinyal *hidden* layer dengan menggunakan persamaan :

$$z_j = f\left(z_{in_j}\right)$$

Dimana :
z_j = nilai aktivasi pada hidden layer
$f(z_{in_j})$ = fungsi sigmoid binner dengan f(x) dari nilai input pada hidden layer
Langkah 3 : menghitung nilai dari hidden layer menuju output layer dengan menggunakan persamaan :

$$y_in_k = w0_k + \sum_{i=1}^{n} z_j\, w_{jk}$$

Dimana :
y_in_k = nilai dari hidden layer menuju output layer
$w0_k$ = bias pada hidden layer menuju output layer
z_j = nilai aktivasi pada hidden layer
w_{jk} = bobot pada lapisan output
Langkah 4 : menghitung fungsi aktivasi pada output layer dengan menggunakan persamaan

$$y_k = f(y_in_k)$$

Dimana :
y_k = nilai aktivasi pada output layer
$f(y_in_k)$ = fungsi *sigmoid binner* dengan f (x) menggunakan nilai dari *hidden layer* ke *output layer*
Langkah 5 : pengecekan nilai *error* pada output layer dengan menggunakan persamaan

$$error_k = t_k - y_k$$

Dimana :
$error_k$= nilai *error* pada output layer
t_k = nilai target keluaran
y_k = nilai aktivasi pada *output layer*
Langkah 6 : menghitung nilai kuadrat *error* dengan persamaan

Kuadrat *error* = $error_k{}^2$

Dimana :
Kuadrat *error* = nilai kuadrat *error* pada output layer
$error_k{}^2$ = nilai error pada output dipangkatkan dua *Backpropagation*
Langkah 7 : menghitung informasi *error* pada *output layer* dengan persamaan

$$\delta_k = (t_k - y_k) f'(y_in_k)$$

Dimana :

δ_k = nilai informasi *error* pada *output layer*

$(t_k - y_k)$ = nilai *error* pada *output layer*

$f'(y_in_k)$ = nilai turunan dari persamaan *sigmoid binner* dengan f (x) dari nilai *hidden layer* menuju *output layer*

Langkah 8 : menghitung koreksi bobot untuk memperbaiki nilai w_{jk} dengan persamaan

$$\Delta w_{jk} = \alpha * \delta_k * z_j$$

Dimana :

Δw_{jk} = nilai koreksi pada *hidden layer* ke *output layer*

α = *learning rate*

δ_k = nilai informasi *error* pada *output layer*

Langkah 9 : menghitung koreksi bias untk memperbaiki nilai w_{0k} dengan persamaan

$$\Delta w_{0k} = \alpha * \delta_k$$

Dimana :

Δw_{0k} = nilai koreksi bias pada *output layer*

α = *learning rate*

δ_k = nilai informasi *error* pada *output layer*

Langkah 10 : menghitung koreksi dari *output layer* menuju *hidden layer* dengan menggunakan persamaan

$$\delta_in_j = \sum_{k=1}^{m} \delta_k w_{kj}$$

Dimana :

δ_in_j = nilai koreksi dari *output layer* menuju *hidden layer*

δ_k = nilai informasi *error* pada *output layer*

w_{kj} = *nilai hidden* layer ke *output layer*

Langkah 11 : menghitung koreksi informasi *error* dengan menggunakan aktivasi pada *hidden layer* dengan menggunakan persamaan

$$\delta_j = \delta_in_j * f'(z_in_j)$$

Dimana :

δ_j = koreksi informasi *error* pada *hidden layer*

δ_in_j = nilaikoreksi *output layer* menuju *hidden layer*

$f'(z_in_j)$ = nilai turunan *sigmoid binner* dengan menggunakan f (x) nilai dari
input menuju ke *hidden layer*

Langkah 12 : menghitung koreksi bobot untuk memperbaiki nilai v_{ij} dengan persamaan

$$\Delta v_{jk} = \propto \delta_j x_i$$

Dimana :

Δv_{jk} = koreksi bobot dari *hidden layer* ke *output layer*

$\propto$ = *learning rate*

δ_j = koreksi informasi pada *hidden layer*

x_i = bobot awal pada *input layer*

Langkah 13 : menghitung koreksi bias dari *input layer* menuju *hidden layer* dengan menggunakan persamaan

$$\Delta v_{0j} = \propto \delta_j$$

Dimana :

Δv_{0j} = koreksi nilai bias dari input ke *hidden layer*

$\propto$ = *learning rate*

δ_j = koreksi informasi pada *hidden layer*

Langkah 14 : menghitung perbaikan nilai bias dan bobot tiap – tiap output dengan menggunakan persamaan

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

Dimana :

$w_{jk}\,(baru)$ = nilai bobot baru pada *hidden* ke *output layer*

$w_{jk}\,(lama)$ = nilai bobot (2.17) *dden* ke *output layer*

Δw_{jk} = koreksi bilai bobot pada *hidden* ke *output layer*

Langkah 15 : menghitung perbaikan bias dari *output* ke *hidden layer* dengan menggunakan persamaan

$$v_{ij}\,(baru) = v_{ij}\,(lama) + \Delta v_{ij}$$

Dimana :

$v_{ij}\,(baru)$ = nilai bias baru dari *input* ke *hidden layer* (2.18)

$v_{ij}\,(lama)$ = nilai bias *input* ke *hidden layer*

Δv_{ij} = nilai koreksi bias dari *inputlayer* ke *hidden layer*

2. Untuk pengujian pada jaringan *Backpropagation* akan melalui tahap *feedforward* saja yaitu den (2.19)

Langkah 1 : menghitung nila apat dari sinyal input menuju *kehidden layer* dengan menggunakan persamaan :

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

Dimana :

z_in_j = nilai dari *input* menuju (2.20) *r*

$v0_j$ = bobot awal bias *in* *hidden layer*dari data latih

x_i = nilai *input layer* dari data uji

v_{ij} = nilai bobot awal *hidden layer* ke *hidden layer*dari data latih

Langkah 2 : menghitung fungsi aktivasi pada sinyal *hidden layer* dengan menggunakan persamaan :

$$z_j = f\left(z_{in_j}\right)$$

(2.21)

Dimana :

z_j = nilai aktivasi pada *hidden layer*

$f(z_{in_j})$ = fungsi *sigmoid binner* dengan f(x) dari nilai input pada *hidden layer*

Langkah 3 : menghitung nilai dari *hidden layer* menuju *output layer* dengan menggunakan persamaan :

$$y_in_k = w0_k + \sum_{i=}^{n} z_i w_{ik}$$

(2.22)

Dimana :

y_in_k = nilai dari *hidden layer* menuju *output layer*

$w0_k$ = bias pada *hidden layer* menuju *output layer*dari data latih
z_j = nilai aktivasi pada *hidden layer*
w_{jk} = bobot pada lapisan *output*dari data latih
Langkah 4 : menghitung fungsi aktivasi pada *output layer* dengan menggunakan persamaan

$$y_k = f(y_in_k)$$

Dimana :
y_k = nilai aktivasi pada *output layer*
$f(y_in_k)$= fungsi *sigmoid binner* dengan f (x) menggunakan nilai dari *hidden layer* ke *output layer*
Langkah 5 : pengecekan nilai *error* pada *output layer* dengan menggunakan persamaan

$$error_k = t_k - y_k$$

Dimana :
$error_k$ = nilai *error* pada *output layer*
t_k = nilai target keluaran
y_k = nilai aktivasi pada *output layer*
Langkah 6 : penentuan klasifikasi dengan membandingkan hasil *error* dengan nilai *thershold*

2.2 Algoritma Radial Basis Function
Tahap pembelajaran tak terawasi
Untuk klasifikasi data pada tahap pembelajaran tak terawasi ini digunakan algoritma K – means. Pada tahap ini akan diapatkan pusat center cluster yang disebut juga center, dari center – center tersebut akan didapatkan lebar (widht) cluster. Berikut tahapan algoritma K – means.

1. Tentukan jumlah pusat cluster
2. Tentukan pusat cluster secara acak
3. Hitung jarak setiap data yang ada terhadap setiap pusat cluster.
   Distence space atau jarak antar dua titik dihitung menggunakan rumus Eucledian yang didefinisikan :

$$D_{12}(x_2 - x_1) = \|x_2 - x_1\| = \sqrt{\sum_{j=1}^{p} (x_{2j} x_{1j})^2}$$

   Dimana : p = dimensi data
4. Kelompokan data berdasarkan jarak minimum terhadap pusat cluster
5. Hitung pusat cluster yang baru dengan cara mengelompokan data bersdasarkan jarak minimum. Dari setiap pusat cluster. Perhitungan pusat cluster yang baru didapatkan menggunakan persamaan :

$$v_i = \frac{\sum_{j=1}^{ni} x_j}{n_i}$$

   Dimana : vi = pisat cluster dari cluster ke – i
   Xj = data masukkan yang merupakan kelompok ke – i
   Ni = banyaknya data kelompok ke – i
6. Ulangi langkah ke – 2 hingga ke – 5 sampai tidak ada perubahan kelompok data berdasarkan jarak minimum terhadap pusat cluster

Setelah menentukan pusat cluster di atas, dapat dilanjutkan dnegan menentukan lebar cluster. Berikut adalah rumus untuk menentukan lebar cluster :

$$r = \frac{d_{max}}{\sqrt{jumlah\ center}} \quad (2.13)$$

Dimana d_{max} adalah jarak m ari jarak setiap pusat cluster. Berikut ac mencari d_{max} :

$$d_{max} = \sqrt{(v_i - v_j)^2} \quad (2.14)$$

Tahap Pembelajaran Terawasi
Bobot pada proses pembelaj ı Radial Basis Function ini diperc i tahap pembelajaran terawasi. Pada tahp ini akan dilakukan perbaikan bobot, dimana data keluaran sudah ditentukan (data aktual). Perbaikan bobot akan menentukan keluaran dari Fungsi Radial Basis. Berikut adalah proses tahap pembelajaran dalam menentukan bobot:

a. Untuk setiap vektor himpunana pembelajaran, ualangi langkah 1 -3
1. Ambil vektor masukan xi dari himpunana pembelajaran
2. Hitung keluaran dari neuron lapisan tersembunyi secara bersamaan ditunjukkan sebagai vektor h

$$h_j = exp\left[\frac{\sum_{i=1}^{n}(x_i - c_j)^2}{2r^2}\right]$$

   Dimana : x = data
   c = center
   r = lebar
3. Hitung vektor keluaran jaringan y. Berikut persamaan mencari y :

Bentuk umum dari RBF adalah :

$$y = f(x) = \sum_{i=1}^{n} w_i h_j$$

b. Hitung error antara hasil keluaran dengan target. Ulangi langkah 1 – 4 samapai batas yang ditentukan sudah dilampaui.
4. Untuk perbaikan bobot, bandingkan dengan vektor target t dengan y. Sesauikan setiap bobot w pada satu arah sehingga mengurangi perbedaan.

Berikut adalah gradient descent algorithm :

$$w_j(n+1) = w_j(n) + \propto (t_i - y_i) h_j$$

Dimana :
wij = bobot antar neuron lapisan tersembunyi i dan neuron lapisan output
$\propto$ = learning rate
tj = target atau output yang diinginkan dari neuron j pada lapisan output
yj = output neuron j pada lapisan output
hj = output lapisan tersembunyi

3. PENUTUP

Berdasarkan rencana pengujian 1 dan 2 dapat ditarik kesimpulan bahwa nilai parameter terbaik untuk metode *backpropagation* dan *radial basis function* dalam menentukan klasifikasi penyakit diabetes mellitus adalah parameter *learning rate* sebesar 0,1 dan parameter epoch sebesar 100. Pada pengujian *cross validation* dapat disimpulkan bahwa metode *radial basis function* memiliki keakurasian yang baik dibandingkan dengan metode *backpropagation* dalam mengklasifikasikan penyakit diabetes mellitus. Dengan nilai akurasi yang diperoleh oleh metode *radial basis function* sebesar 67,5% sedangkan nilai akurasiuntuk metode *backpropagation* sebesar 60%. Oleh karena itu dalam kasus klasifikasi penyakit diabetes mellitus dengan menggunakan metode *backpropagation* dan *radial basis function* dapat disimpulkan bahwa metode *radial basis function* lebih baik dibandingkan metode *backpropagation* dalam kasus klasifikasi penyakit diabetes mellitus.

DAFTAR PUSTAKA

[1] Fajar AuliaRachman. "Diteksi Penyakiut Kulit Mengguakan Filter 2D Gabor Wavelevet dan Jaringan Saraf Tiruan Radial Basis Function (RBF)". [Diakses 3 september 2015]

[2] Novi Indah Pradasari. "Aplikasi Jaringan Syaraf Tituan Memprediksi Penyakit Saluran Pernafasan dengan Metode Backpropagation". [Diakses 3 september 2015]

[3] "Jaringan Syaraf Tiruan" diyah puspita nigrum , C.V andi offset 2006

[4] "Artificial Intelegent " sri kusumadei , graha ilmu, 2003

[5] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," International Joint Conference on Artificial Intelligence (IJCAI), 1995.

[6] H.Hamilton, "Confussion Matrix," [Online]. Available:http:// www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html. [Diakses 29 December 2015].

[7] A.S. Rosa dan M. Shalahuddin, Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek), Bandung: Modula, 2013.

[8] Hartanto, B., "Memahami Visual C#.Net Secara Mudah". Yogyakarta : Andi, 2008.

[9] Arif jumarwanto."Aplikasi Jaringan Syaraf Tiruan Backpropagation untuk Memprediksi penyakit THT di rumah sakit rahayu kudus"[Diakses 8 oktober 2015]

[10] Sri redjeki."Perbandingan Algoritma Backpropagation dan K- Nearst untuk identifikasi penyakit".[Diakses 8 oktober 2015]

[11] "Pharamaceutical care untuk penyakit diabetes mellitus" direktorat bina farmasi konumitas dan klinik direktorat jendaral bina kefarmasian dan alat kesehatan departemen kesehatan RI.[Diakses 9 september]