PERFORMANCE COMPARISON BACKPROPAGATION METHOD AND RADIAL BASIS FUNCTION (RBF) FOR CLASSIFICATION PROBLEMS DIABETES MELLITUS DISEASE

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

ABSTRACT

Artificial neural network is one method that can be used to classify a data.Metode artificial neural networks are often used to classify diseases including Backpropagation and Radial Basis Function. Backpropagation use error values to change the weight - the weight in two directions, namely toward backward and forward so that it gets a good output. Radial Basis Function undergo two stages of learning with supervised and unsupervised. Diabetes is a common disease suffered by the community. Diabetes is divided into two types of diabetes mellitus insulin and non-insulin diabetes mellitus. Handling each - each type of diabetes is different - different, the need for examination early age so that every patient with diabetes mellitus had been treated in accordance with the type of diabetes in the suffering.
Based on the results of tests performed superbly radialbasis method function has a value which is less than the error propagation method.

Keywords: Backpropagation, Radial Basis Function, k-fold cross validation

1. PRELIMINARY

Identifying characteristics of an object is one way to describe or classify a particular group of data based on similar characteristics that exist in an object. How to perform a classification one can use the method of artificial neural networks. Artificial neural network method that is often used to classify diseases including Backpropagation and Radial Basis Function. Both methods have respective characteristics - each within the classification. Backpropagation use error values to change the weight - the weight in two directions, namely toward backward and forward so that it gets a good output. Radial Basis Function undergo two stages of learning with supervised and unsupervised, unchecked mean there are other algorithms to aid in the learning process to determine the needs in the learning process and perform radial basis function algorithm for further processing in the learning process.
Since the method of Backpropagation and Radial Basis Function is often used for in classification of disease. Diabetes mellitus require classification in order to conduct follow-up obtained the appropriate treatment based on the classification of types of diabetes mellitus.

1.1 Backpropagation Neural Network

Backpropagation Neural Network is a supervised learning algorithm and is typically used by perceptron with many layers to change the weight - weight that is connected to neurons - neurons that exist in the hidden layer. Algorithm backpropagation neural netwrok using the error output to change the value of weight - weight in a backward direction (backward). To get this error, advanced propagation phase (forward propagation) must be done first.

1.2 Radial Basis Function

Radial Basis Function Network is one example of a neural network with many layers to the direction of propagation of advanced learning (Multilayer feedforward neural). The network has three pieces of coating the layer of input, hidden and output. In the Radial Basis Function, there is only one hidden layer pieces only.
Radial Basis Function is memiliji value maksismum 1, which occurs when the received input value is 0 (the distance between the weights with input 0, so that when the distance between the weights with input berkuran, this function will generate greater output. Umunya activation function Radial Basis Function use Gaussian and linear function of the output layer. Here tends Gaussian activation function:

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

Where :
x = input
ci = data centers to - i, i = 1, 2, ..., n
$\sigma_j{}^2$ = standard deviation to - j, j = 1, 2, ..., n
n = many data centers

2.3 Accuracy Testing

To measure a test can be done by calculating the difference between the data calculation results with actual destination data. Difference in magnitude (size error testing) the test data to the actual data that happened a penring of data to assess the accuracy of a test method. By comparing the size of the error several methods of testing, will be obtained which method has the size of the smallest error, so that the value of the test can be used as a reference in determining the need - a need which is to come. There are some errors in the size of the testing, among others.

Root Mean Squared Error (RMSE)

RMSE is the square root of the mean value error with the following equation:

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

RMSE is an alternative method to evaluate the forecasting techniques. RMSE is the average - average square of the difference value of the estimated value of a variable observation. If the value of the smaller RMSE then estimate the model or the variable semakain valid. Notation basis of the above equation are summarized as follows:

Xt = actual value in period t
Ft = value testing in period t
Xt - Ft = value testing error
n = number of data

2. RESEARCH

In this case the two methods among the methods Backpropagation and Radial Basis Function measing - each doing the learning and testing were done for performance comparison tuajuan know which one is better.

2.1 Backpropagation Algorithm Method

The algorithm method BackpropagationNeural Network:

Assigning weights (w), learning rate (α), the maximum epoch (MaxEpoch).
Determining the weight initialization (take a random value that is small enough)
The initial weight input to the hidden layer [v_{ij}]
The initial weight bias to the hidden layer [v_{0j}]
The initial weight hidden layer to the output layer [w_j]
The initial weight bias to the output layer [w_0]
Work steps - these steps during a stop condition appreciating FALSE:
For each -tiap input layer that will do the learning, working on:
feedforward:
Step 1: calculate the value derived from the input signal to the kehidden layer using the equation:

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

Where :
z_in_j *= the value of the input to the hidden layer*
$v0_j$ *= initial weight bias input to the hidden layer*
x_i *= the value of the input layer;*
x1 = height
x2 = weight
x3 = tension
x4 = blood sugar before the fasting (GDP)
x5 = blood sugar after fasting (GDPP)
v_{ij} *= initial weight value hidden layer to the hidden layer*
Step 2: calculate the activation function in the hidden layer signals using the equation:

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

Where :
z_j = activation value in the hidden layer
$f(z_{in_j})$ *= Binner with the sigmoid function f (x) of the input values in the hidden layer*
Step 3: calculate the value of the hidden layer to the output layer using the equation:

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

Where :
y_in_k *= the value of the hidden layer to the output layer*
$w0_k$ *= bias in the hidden layer to the output layer*
z_j *= activation value in the hidden layer*
w_{jk} *= weights in the output layer*
Step 4: calculate the activation function in the output layer using the equation

$$y_k = f(y_in_k)$$

Where :
y_k *= value at the output layer activation*
$f(y_in_k)$*= Binner with the sigmoid function f (x) using the value of the hidden layer to the output layer*
Step 5: checking the value of the error in the output layer using the equation

$$error_k = t_k - y_k$$

Where :
$error_k$ *= value error at the output layer*
t_k *= target value output*
y_k *= value at the output layer activation*
Step 6: calculate the value of the square error with equations
Squared error = error
Where :
Squared error = $error_k{}^2$
Squared error = value error at the output raised to a power of two
backpropagation
Step 7: calculate the error information in the output layer to the equation

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

Where :
δ_k *= nilai informasi error pada output layer*
$(t_k - y_k)$ *= nilai error pada output layer*

$f'(y_in_k)$ = the value of the derivative of the sigmoid equation Binner with f (x) of the hidden layer to the output layer

Step 8: calculate the correction weight to correct the value wjk with equation

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

Where :

Δw_{jk} = corrected value of the hidden layer to the output layer

α = learning rate

δ_k = the value of the error information in the output layer

Step 9: calculate the bias correction remedy fix w0k value to the equation

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

Where :

Δw_{0k} = the value of the refractive correction on the output layer

α = learning rate

δ_k = the value of the error information in the output layer

Step 10: calculate the correction of the output layer to the hidden layer using the equation

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

Where :

δ_in_j = corrected value of the output layer to the hidden layer

δ_k = the value of the error information in the output layer

w_{kj} = value hidden layer to the output layer

Step 11: calculate error correction information by using the activation of the hidden layer using the equation

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

Where :

δ_j = correction of an error in hidden layer information

δ_in_j = nilaikoreksi output layer to the hidden layer

$f'(z_in_j)$ = value of derivative sigmoid Binner using f (x) the value of

input leading to the hidden layer

Step 12: calculate the correction weights to improve the value equation vijdengan

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

Where :

Δv_{jk} = correction weights from the hidden layer to the output layer

α = learning rate

δ_j = correction information in the hidden layer

x_i = initial weight in the input layer

Step 13: calculate the refractive correction of the input layer to the hidden layer using the equation

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

Where :

Δv_{0j} = correction bias value from the input to the hidden layer

α = learning rate

δ_j = correction information in the hidden layer

Step 14: calculate the bias value improvement and weight of each - each output using the equation

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

Where :

$w_{jk}\,(baru)$ = new weight value on a hidden layer to output

$w_{jk}\,(lama)$ = the value of the old weights of the hidden layer to output

Δw_{jk} = correction weight to the hidden layer to output

Step 15: calculate the bias improvement of output to hidden layer using the equation

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

Where :

$v_{ij}\,(baru)$ = new bias value from the input to the hidden layer

$v_{ij}\,(lama)$ = the value of the old bias from input to hidden layer

Δv_{ij} = value bias correction of inputlayer to the hidden layer

For testing the Backpropagation network feedforward just going through a phase that is through

Step 1: calculate the value derived from the input signal to the kehidden layer using the equation:

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

Where :

z_in_j = the value of the input to the hidden layer

$v0_j$ = initial weight bias towards the hidden layerdari input training data

x_i = the value of the input layer of the test data

v_{ij} = initial weight value hidden layer to the hidden layerdari training data

Step 2: calculate the activation function in the hidden layer signals using the equation:

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

Where :

z_j = activation value in the hidden layer

$f(z_{in_j})$ = Binner with the sigmoid function f (x) of the input values in the hidden layer

Step 3: calculate the value of the hidden layer to the output layer using the equation:

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

Where :

y_in_k = the value of the hidden layer to the output layer

$w0_k$ = bias in the hidden layer to the output layerdari training data

w_{jk} = weight of the lining of training data outputdari

Step 4: calculate the activation function in the output layer using the equation

$y_k = f(y_in_k)$

Where :

y_k = value at the output layer activation

$f(y_in_k)$= *Binner with the sigmoid function f (x) using the value of the hidden layer to the output layer*
Step 5: checking the value of the error in the output layer using the equation
$error_k = t_k - y_k$
Where :
$error_k$ *= value error at the output layer*
t_k *= target value output*
y_k *= value at the output layer activation*
Step 6: determination of the classification error by comparing the results with values thershold

2.2 Algorithm Radial Basis Function
Unsupervised learning phase
For the classification of data at this stage unsupervised learning algorithm, K - means. At this stage it will diapatkan center center cluster also called the center, of the center - the center will be obtained width (widht) cluster. The following stages of the algorithm K - means.
Determine the number of cluster centers
Specify the cluster centers randomly
Calculate the distance of each data available to each cluster center.
Distence space or distance between two points is calculated using the formula defined Eucledian:
$D_{12}(x_2 - x_1) = \|x_2 - x_1\| = \sqrt{\sum_{j=1}^{p}(x_{2j}\, x_{1j})^2}$
Where: p = dimension of data
Group data based on the minimum distance to the cluster center
Calculate new cluster centers with a way to classify the data bersdasarkan minimum distance. From each cluster center. Calculation of a new center cluster obtained using the equation:
$v_i = \frac{\sum_{j=1}^{ni} x_j}{n_i}$
Where: vi = pisat from cluster to cluster - i
Xj = Data input which is a group to - i
Ni = the number of data groups to - i
Repeat steps - up to 2 - 5 until no change in the data group based on the minimum distance to the cluster center
After determining the center cluster above, can be continued circuitry determines the width of the cluster. Here is the formula to determine the width of the cluster:
$r = \frac{d_{max}}{\sqrt{jumla\ h\ center}}$
Where d_max is the maximum distance from the center within each cluster. Here is the formula to find d_max:

$d_{max} = \sqrt{(v_i - v_j)^2}$
Supervised Learning Stage
The weights in the learning process Radial Basis Function network is obtained through supervised learning phase. In this tahp will repair the weights, where the output data are set (actual data). Repair weight will determine the output of Radial Basis Function. Here is the learning stage process in determining the weights:
For each vector himpunana learning, ualangi step 1 -3
Take the input vector xi of learning himpunana
Calculate the output of the hidden layer neurons simultaneously shown as vectors h
$h_j = exp\left[\frac{\sum_{i=1}^{n}(x_i - c_j)^2}{2r^2}\right]$
Where: x = Data
c = center
r = width
Calculate the network output vector y. The following equations are looking for y:
The general form of the RBF is:
$y = f(x) = \sum_{i=1}^{n} w_i h_j$
Calculate the error between the output target. Repeat steps 1-4 samapai prescribed limit has been exceeded.
For the improvement of weight, compared with a target vector t y. Sesauikan each weight w in one direction, thereby reducing the difference.

Here is a gradient descent algorithm:
$w_j(n+1) = w_j(n) + \propto (t_i - y_i)h_j$
Where :
wij = weights between the hidden layer neuron i and neuron output layer
α *= learning rate*
tj = target or desired output of neuron j in the output layer
yj = output neuron j in the output layer
hj = output layer is hidden

3. CLOSING

Under the plan the tests 1 and 2 can be deduced that the best parameter values for the method of backpropagation and radial basis function in determining the classification of diabetes mellitus is the learning rate parameter of 0.1 and parameter epoch of 100. In cross validation test can be concluded that the method of radial basis function has good accuracy compared with propagation method in classifying diabetes mellitus. With the value of accuracy obtained by the method of radial basis function of 67.5% while the value akurasiuntuk method of propagation by 60%. Therefore, in the case of the classification of diabetes mellitus using backpropagation and radial basis function can be concluded that the radial basis function method is better than the method of propagation in the case of the classification of diabetes mellitus.

BIBLIOGRAPHY

[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]