Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

Comparison between Neural Network (NN) and Adaptive Neuro Fuzzy Inference System (ANFIS) on sunlight intensity prediction based on air temperature and humidity

To cite this article: D Rahmalia *et al* 2020 *J. Phys.: Conf. Ser.* **1538** 012044

Comparison between Neural Network (NN) and Adaptive Neuro Fuzzy Inference System (ANFIS) on sunlight intensity prediction based on air temperature and humidity

D Rahmalia[1], T Herlambang[2], A S Kamil[2], R A Rasyid[3], F Yudianto[2], L Muzdalifah[4], E F Kurniawati[4]

[1]Mathematics Department, University of Islam Darul Ulum Lamongan
[2]Information System Department, University of Nahdlatul Ulama Surabaya
[3]Management Department, University of Nahdlatul Ulama Surabaya
[4]Mathematics Department, University of PGRI Ronggolawe Tuban

Email :teguh@unusa.ac.id, dinitarahmalia@gmail.com

Abstract. Weather prediction expecially in predicting sunlight intensityhas important role in energy usage. As effort for controlling petrol-based fuel usage, government manage energy usage by converting solar energy from sunlight intensity to electric through solar cell.Sunlight intensity depends on air temperature and humidity. Two methods on prediction process will be applied : Neural Network (NN) and Adaptive Neuro Fuzzy Inference System (ANFIS). Type of NN used in prediction process is Backpropagation. Backpropagation consists of forward propagation, backward propagation, and update weight matrices. ANFIS uses hybrid method to train consequent parameters and premise parameters. In this research, NN will be compared with ANFIS. From five trials of NN simulations, the number of maximum epoch for making the root of mean square error(RMSE) in training data is very large so that the computation time is very long. From the comparison result of two methods, we can see that ANFIS can make faster prediction than NN with the number of maximum epoches is smaller than NN so that computation time is faster.

1. Introduction

Weather prediction is important work for some activities. In weather prediction, there are many conditions which can be observed such as air temperature, humidity, sunlight intensity, and so on. Weather prediction expecially in determining sunlight intensity is important work in energy usage. According data from Energy and Mineral Resources Ministry during year 2005-2010, the number of consumed fuel in Indonesia is more than the number of produced fuel. This problem causes government to import the fuel. As effort for controlling petrol-based fuel usage, government manage energy usage by converting solar energy from sunlight intensity to electric through solar cell [13]. This research will compare two methods for weather prediction, such as Neural Network (NN) and Adaptive Neuro Fuzzy Inference System (ANFIS).

Neural Network (NN) was introduced by McCulloch and Pitts in 1943. NN work resembles to human neuron system. Type of NN used in prediction process is Backpropagation. Backpropagation consists of forward propagation, backward propagation, and update weight matrices. In forward propagation, some calculations using activation function start from input, hidden layer, and output respectively. In backward propagation, error factor calculations are applied from output, hidden layer, and input respectively. After that, update weight matrices [4].

The applications of fuzzy logic have been applied, such as for clustering the data [8] and forecasting [1,2]. Adaptive Neuro Fuzzy Inference System (ANFIS)is the network which uses supervised learning algorithm. It has a similar function to the Takagi-Sugeno fuzzy inference system model. In addition, adaptive network has the architecture characteristics that consists a number of adaptive nodes. Every node has different functions and the output is affected from the signals and parameters [2]. ANFIS uses hybrid method to train consequent parameters and premise parameters [1].

In previous research, forecasting methods for prediction have been applied by exponential smooting [5], Kalman Filter [6,7]. NN expeciallyBackpropagation has been applied in weather prediction [9,10,12]. This algorithm is applied in training data and testing data with certain proportion [11]. In this research, Backpropagation will be compared with ANFIS. From five trials of NN simulations, the number of maximum epoch for making the root of mean square error (RMSE) in training data is very large so that the computation time is very long. From the comparison result of two methods, we can see that ANFIS can make faster prediction than NN with the number of maximum epoches is smaller than NN so that computation time is faster.

2. Neural Network

Neural Network (NN) was introduced by Mc Culloch and Pitts in 1943. NN works resemble to human's neuron like :

a. The signal is travelled between neuron through connector.
b. Connectors have weight which will either increase or decrease signal.
c. To determine output, neuron uses activation function applied in the sum of input received.

In NN, activation function is used for determining output of neuron. Argument of activation function is linear combination of input and weight as in equation (1).

$$\begin{aligned} net &= \sum_i x_i w_i \\ f(net) &= f\left(\sum_i x_i w_i\right) \end{aligned} \tag{1}$$

Generally, the properties of activation function are : continuous, differentiable, and not descending function (Fausett, 1994). In this research, activation function applied is binary sigmoid with range (0-1) as in equation (2).

$$\begin{aligned} f(net) &= \frac{1}{1+e^{-net}} \\ f'(net) &= f(net)(1-f(net)) \end{aligned} \tag{2}$$

2.1. Backpropagation Structure

Backpropagation is type of NN used in prediction process. Backpropagation consists some inputs $x_1, x_2, ..., x_n$, some hidden layer $z_1, z_2, ..., z_p$, and some output $y_1, y_2, ..., y_m$. In input and hidden layer, there is bias having value 1.Weight v_{ij} connects input x_i to hidden layer z_j. Weight w_{jk} connects hidden layer z_j to output y_k. In Backpropagation, there are three phases of calculation : forward propagation, backward propagation, and update weight matrices [4]. Backpropagation model can be seen on figure 1.

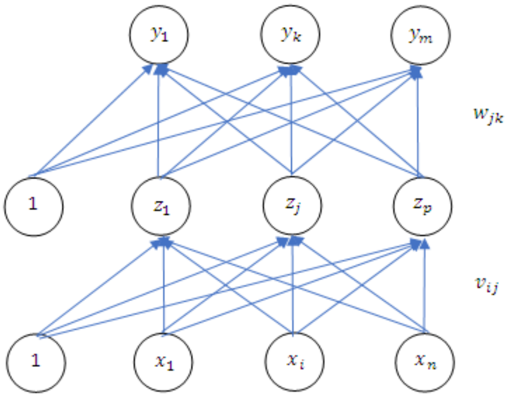


Figure 1. Backpropagation model

2.2. Backpropagation Algorithm

The algorithm of Backpropagation can be explained as follows :

Initialization weight matrices V and W with random number between -0.5 to 0.5.

$e = 1$

$while(e \le max\_epoch \,\&\&\, MSE \ge min\_MSE)$

$for(d = 1 : datasize)$

1. For each input receives signal and continues to each hidden layer through forward propagation in equation (3) until equation (6) and backward propagation in equation (7) until equation (11).
   Forward Propagation
2. Calculating all outputs in hidden layer $z_j, j = 1,2,.., p$

$$z\_net_j = v_{oj} + \sum_{i=1}^{n} x_i v_{ij} \tag{3}$$

$$z_j = f(z\_net_j) = \frac{1}{1 + e^{-z\_net_j}} \tag{4}$$

3. Calculating all outputs in output $y_k, k = 1,2,..., m$

$$y\_net_k = w_{0k} + \sum_{j=1}^{p} z_j w_{jk} \tag{5}$$

$$y_k = f(y\_net_k) = \frac{1}{1 + e^{-y\_net_k}} \tag{6}$$

   Backward Propagation
4. Calculating factor δ output based on error in each output $y_k, k = 1,2,..., m$

$$\delta_k = (t_k - y_k) f'(y\_net_k) \tag{7}$$

5. Calculating weight update

$$\Delta w_{jk} = \alpha \delta_k z_j, \quad k = 1,2,..,m \;\; j = 0,1,2,..,p \tag{8}$$

6. Calculating factor δ hidden layer based on error in each hidden layer $z_j, j = 1,2,...,p$

$$\delta\_net_j = \sum_{k=1}^{m} \delta_k w_{jk} \tag{9}$$

$$\delta_j = \delta\_net_j \, f'(z\_net_j) \tag{10}$$

7. Calculating weight update

$$\Delta v_{ij} = \alpha \delta_j x_i, \quad j = 1,2,...,p \;\; i = 0,1,2,...,n \tag{11}$$

<u>Update Weight Matrices</u>

8. Updating new weight matrices

$$w_{jk} = w_{jk} + \Delta w_{jk} \tag{12}$$

$$v_{ij} = v_{ij} + \Delta v_{ij} \tag{13}$$

end

Computing the root of mean square error(RMSE)

$$RMSE = \sqrt{\frac{1}{datasize} \sum_{d=1}^{datasize} \frac{1}{m} \sum_{k=1}^{m} (T_{dk} - Y_{dk})^2} \tag{14}$$

with T_{dk} istarget valueand Y_{dk} isoutputs.

$e = e + 1$

end

3. Adaptive Neuro Fuzzy Inference System (ANFIS)

Adaptive Neuro Fuzzy Inference System (ANFIS) architecture is the network which uses supervised learning algorithm. It has similar function to the Takagi-Sugeno fuzzy inference system model. Figure 2 shows the diagramof ANFISwhich consists of five layers[1].

Given two inputs x and y, and one output f. Suppose rules applied are :

Rule 1 : If x is A_1 and y is B_1 then $f_1 = p_1 x + q_1 y + r_1$

Rule 2 : If x is A_2 and y is B_2 then $f_2 = p_2 x + q_2 y + r_2$

with A_1, A_2 and B_1, B_2 are membership functions. Parameter p_1, q_1, r_1 and p_2, q_2, r_2 are consequent parameters.

ANFIS is little different from NN in calculation of each layer, while NN has similar calculation in its hidden layers. Each layer of ANFIS has different calculation given in equation (15) until equation (19) as follows :

 1538 (2020) 012044 doi:10.1088/1742-6596/1538/1/012044

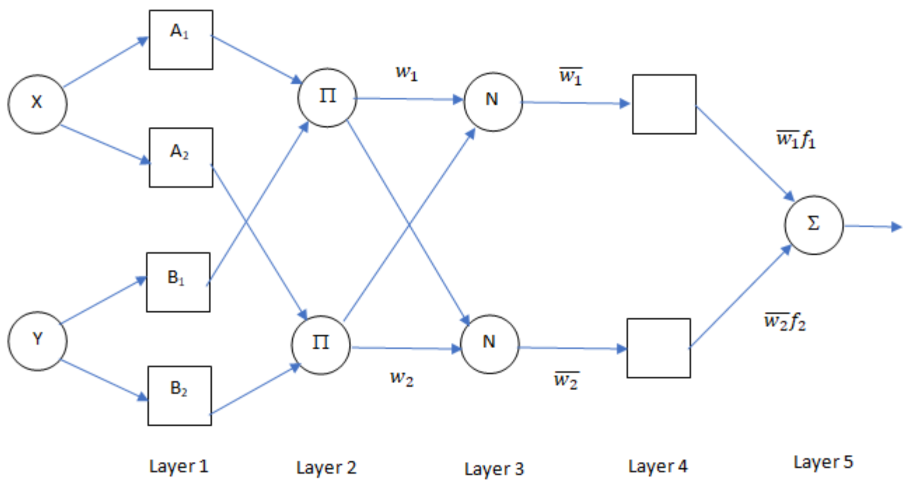


Figure 2. ANFIS structure

Layer 1. In this layer, each node adapts to a function parameter. The output in each node is thedegree of membership valuefrom the membership functions.

$$O_{1,i} = \mu_{Ai}(x)\ i = 1,2 \qquad\qquad O_{1,i} = \mu_{Bi-2}(y)\ i = 3,4 \tag{15}$$

Membership function used is Gaussian membership $\mu(x) = \exp\left(-\left(\frac{x-c}{a}\right)^2\right)$ or generalized bell membership $\mu(x) = \frac{1}{1+\left(\frac{x-c}{a}\right)^{2b}}$. The parameters a, b, c of membership function are premise parameters.

Layer 2. In this layer, every node represents the firing strength from given rule. The output node is the result of multiplying of signal and it is delivered to the next node.

$$O_{2i} = w_i = \mu_{Ai}(x) * \mu_{Bi}(y)\ i = 1,2 \tag{16}$$

Layer 3. Every node is ratio between the i-th firing strength and the total of firing strengths. This result is called the normalized firing strength.

$$O_{3i} = \overline{w_i} = \frac{w_i}{\sum_i w_i}\ i = 1,2 \tag{17}$$

Layer 4. Every node is function defined as :

$$O_{4i} = \overline{w_i} f_i = \overline{w_i}(p_i x + q_i y + r_i)\ i = 1,2 \tag{18}$$

Layer 5. This node sums all signals from the previous nodes.

$$\mathrm{O}_5 = \sum_i \overline{w_i} f_i \tag{19}$$

3.1. Hybrid Method

Hybrid method was proposed by Jang in 1993 to train consequent parameters and premiseparameters. There are two parts of hybrid method : forward path and backward path [4]. In forward path, signals travel until layer 4 and the consequent parameters are computed by the least square estimation (LSE). In backward path, the error rates travel backward and the premise parameters are computed by the gradient descent. One level of hybrid method is called epoch.Hybrid methodcombining LSE and gradient descentcanresultfaster convergencerate than Backpropagation method[1,2].

3.1.1. Optimization on Premise Parameters

Premise parameters a, b, c can be trained and optimized usingBackpropagationandgradient descent as follows :

Suppose networks have L layers and in the k -th , it consists of $\#(k)$ nodes, so that node i of the k -th layer is (k,i) with the output is O_i^k . Supposethe number of training data is p , so thatthe sum of square error E_p is as equation (20).

$$E_p = \sum_{m=1}^{\#(L)} \left(T_{m,p} - O_{m,p}^L\right)^2 \tag{20}$$

with $T_{m,p}$ is target valueof m -th component of p -th data and $O_{m,p}^L$ is output of m -th component in the layer L by p -th data. All computationsfor N training data can be computed by $E = \sum_{p=1}^{N} E_p$

After E_p is determined, compute gradient descent in layer L

$$\frac{\partial E_p}{\partial O_{i,p}^L} = -2\left(T_{i,p} - O_{i,p}^L\right) \tag{21}$$

Generally, in the node (k,i) can be computed using chain rule as in equation (22)

$$\frac{\partial E_p}{\partial O_{i,p}^k} = \sum_{m=1}^{\#(k+1)} \frac{\partial E_p}{\partial O_{m,p}^{k+1}} \cdot \frac{\partial O_{m,p}^{k+1}}{\partial O_{i,p}^k} \tag{22}$$

for each $1 \le i \le \#(k)$ with $1 \le k \le L-1$

If α is the premise parameter of network, then the chain rule is like in equation (23) with S is the set of nodes with outputs depend on α .

$$\frac{\partial E_p}{\partial \alpha} = \sum_{O^* \in S} \frac{\partial E_p}{\partial O^*} \cdot \frac{\partial O^*}{\partial \alpha} \tag{23}$$

All error valuefor N training data can be computed using equation (24).

$$\frac{\partial E}{\partial \alpha} = \sum_{p=1}^{N} \frac{\partial E_p}{\partial \alpha} \tag{24}$$

Update premise parameters usingequation (25) and equation (26).

$$\Delta \alpha = -\eta \frac{\partial E}{\partial \alpha} \tag{25}$$

$$\alpha_{new} = \alpha_{old} - \Delta \alpha = \alpha_{old} - \left(-\eta \frac{\partial E}{\partial \alpha} \right) \tag{26}$$

With η is the learning rate.

3.1.2. Optimization on Consequent Parameters

Optimization on consequent parameters use Least Square Estimation (LSE). Suppse linear combination of output is :

$$\begin{aligned} f &= \bar{w}_1 f_1 + \bar{w}_2 f_2 \\ f &= \bar{w}_1 \left(p_1 x + q_1 y + r_1 \right) + \bar{w}_2 \left(p_2 x + q_2 y + r_2 \right) \\ f &= \left(\bar{w}_1 x \right) p_1 + \left(\bar{w}_1 y \right) q_1 + \left(\bar{w}_1 \right) r_1 + \left(\bar{w}_2 x \right) p_2 + \left(\bar{w}_2 y \right) q_2 + \left(\bar{w}_2 \right) r_2 \end{aligned} \tag{27}$$

For N training data can be constructed to become linear equations system and they can be modified to become matrix system $A\theta = Y$:

$$\begin{bmatrix} (\bar{w}_1 x)_1 & (\bar{w}_1 y)_1 & (\bar{w}_1)_1 & (\bar{w}_2 x)_1 & (\bar{w}_2 y)_1 & (\bar{w}_2 r)_1 \\ (\bar{w}_1 x)_2 & (\bar{w}_1 y)_2 & (\bar{w}_1)_2 & (\bar{w}_2 x)_2 & (\bar{w}_2 y)_2 & (\bar{w}_2)_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ (\bar{w}_1 x)_N & (\bar{w}_1 y)_N & (\bar{w}_1)_N & (\bar{w}_2 x)_N & (\bar{w}_2 y)_N & (\bar{w}_2)_N \end{bmatrix} \begin{bmatrix} p_1 \\ q_1 \\ r_1 \\ p_2 \\ q_2 \\ r_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix} \tag{28}$$

Therefore matrix $\theta = \begin{bmatrix} p_1 & q_1 & r_1 & p_2 & q_2 & r_2 \end{bmatrix}^T$ can be computed byLSE :

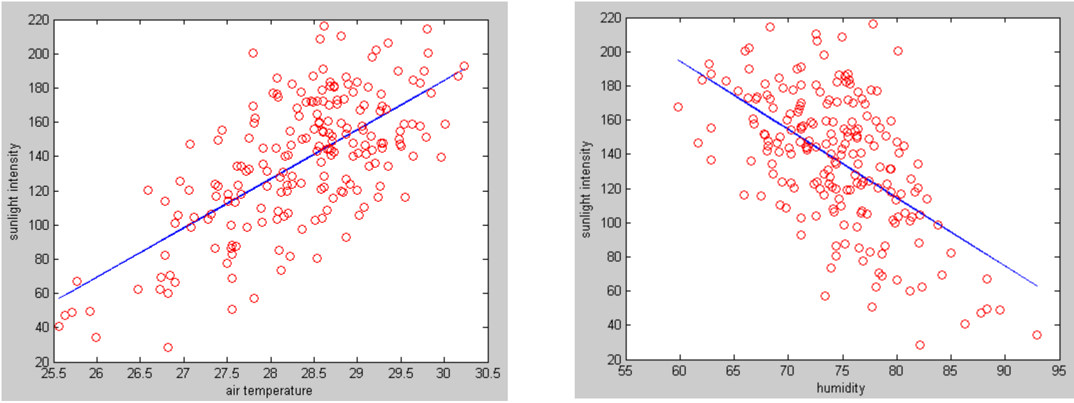
$$\theta = (A^T A)^{-1} A^T Y \tag{29}$$

4. Simulation Results

Dataset are taken from weather condition in one city of Indonesia during January-July 2011 (200 days).The simulations will compare two methods : Neural Network (NN) and Adaptive Neuro Fuzzy Interface System (ANFIS) with the parameter provided. Both NN and ANFIS, we use five trials with NN uses five different initial weights and ANFIS uses five different membership functions.

4.1. The Correlation of Data

The dataset consist information about air temperature (Celcius), humidity (%), and sunlight intensity (Watt/m^2). To determine input and output, we apply data regression and correlation to recognize the relation of data [3]. From the simulation on figure. 3, we obtain that air temperature has positive correlation against sunlight intensity $r_{x_1 y} = 0.676$ while humidity has negative correlation against sunlight intensity $r_{x_2 y} = -0.565$.



(a) (b)

Figure 3. (a) Linear regression of air temperature and sunlight intensity (b) Linear regression of humidity and sunlight intensity

4.2. Data Partition

Before applying prediction process, we need to split the data into training data and testing data. Data partition used is as follows : for training data, data used have proportion 80% of all data while for testing data, data used have remaining proportion (20% of all data) [3]. Figure 4 shows the data partition where red plus marks represent distribution of testing data.

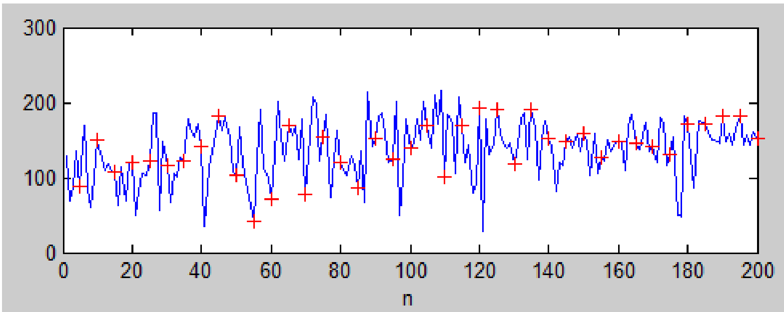


Figure 4. Data partition into training data (blue) and testing data (red)

4.3. The Simulation of Neural Network

Paramaters used in NN (Backpropagation) simulation are :

Learning rate : 0.2
The number of hidden layer : 2
Maximum epoch : 1000
Model of backpropagation : 2(input) – 4 (hidden layer 1) – 4(hidden layer 2) – 1(output)

The simulation of NN can be seen on figure 5, figure 6 and figure 7. First, initialization weight matrices and apply to training data using Backpropagation until maximum epoch and the convergence process can be seen at figure 5. It seems in the early epoch, the RMSE resulted is quite large. In the optimization, the RMSE is gradually decreased and converged. Figure 6 shows simulation on training data. Then, optimal weight matrices are applied to testing data. Figure 7 shows simulation on testing data.

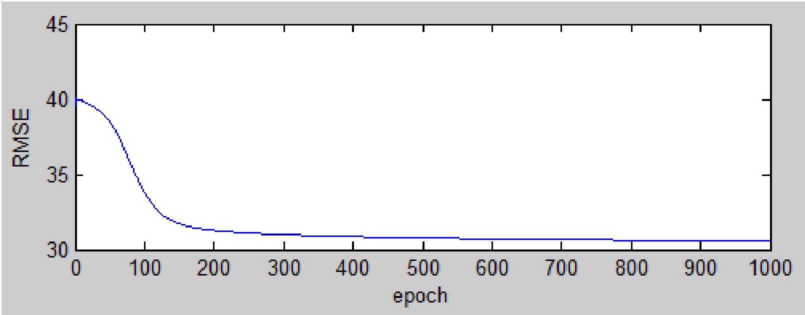


Figure 5. Convergence process on training data using NN (Backpropagation)

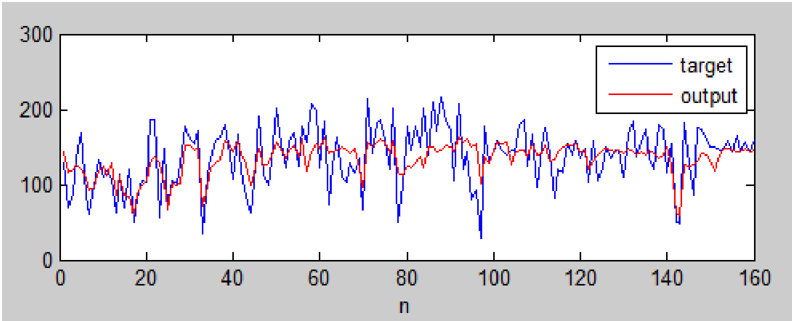


Figure 6. Simulation result using NN (Backpropagation) on training data

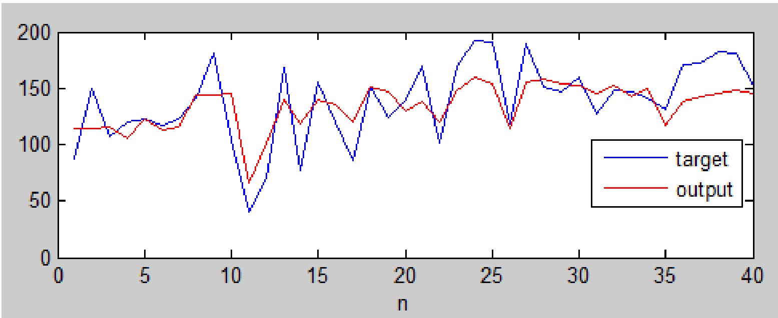


Figure 7. Simulation result using NN (Backpropagation) on testing data

From the simulation, we obtain the prediction with the the root of mean square error (RMSE) are :
Training data : 30.58
Testing data : 23.77

We repeat the simulationsusing different initial weights and the results are in table 1.

Table 1. The root of mean square error (RMSE) on NN (Backpropagation) in five trials

Number	Max. epoch	RMSE on training data	RMSE on testing data
1	1000	30.55	23.73
2	1000	30.67	23.91
3	1000	30.52	23.74
4	1000	30.61	23.81
5	1000	30.47	23.67

4.4. The Simulation of Adaptive Neuro Fuzzy Inference System

Paramaters used in ANFIS simulation are :

Maximum epoch	: 30
The number of membership function	: 3 (input 1), 3 (input 2)
The number of rules	: 9
The number of linear parameter (consequent)	: 27
The number of nonlinear parameter (premise)	: 18

The simulation of ANFIS can be seen on figure 8, figure 9, figure 10 and figure 11. There are three membership functions used where blue curve represents ‘small’, green curve represents ‘medium’, and red curve represents ‘large’. Figure 8 shows optimized membership functions of input air temperature and input humidity. Each of membership function uses generalized bell (gbell) having three parameters as nonlinear parameters. Figure 9 shows convergence process of RMSE. Figure 10 shows prediction result (sunlight intensity) on training data. Figure 11 shows prediction result (sunlight intensity) on training data.

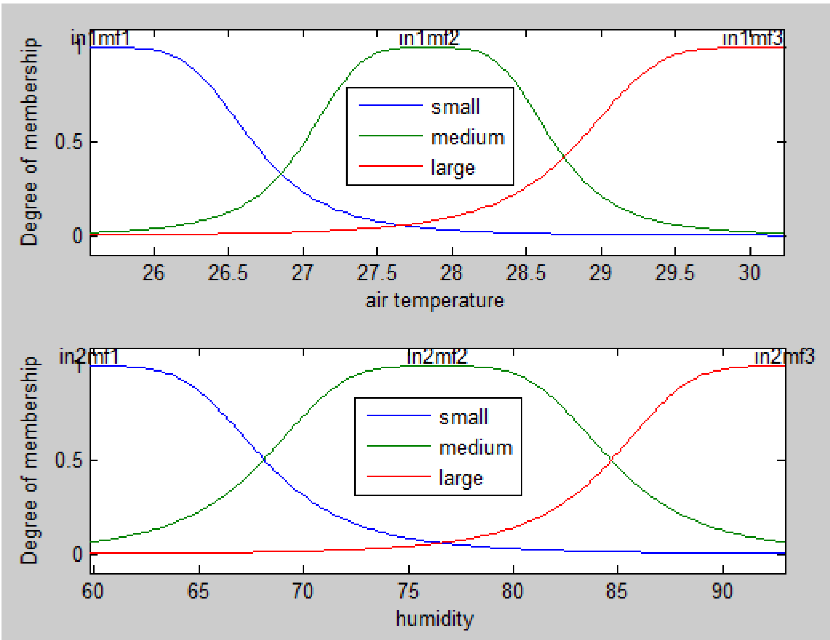


Figure 8. Optimized membership function withgeneralized bell type

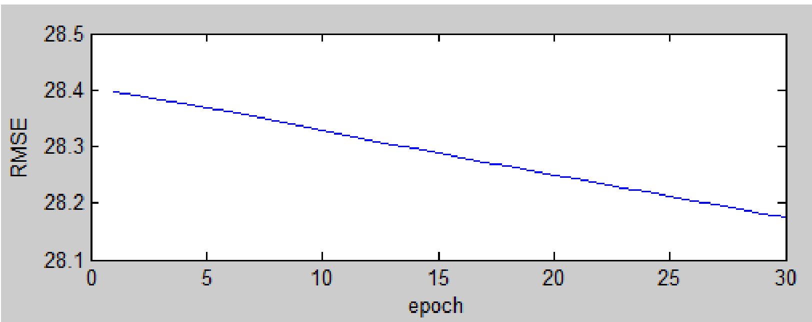


Figure 9. Convergence process on training data using ANFIS

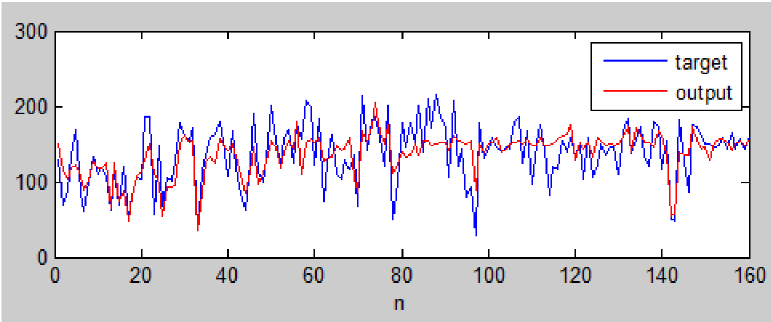


Figure 10. Prediction result using ANFIS on training data

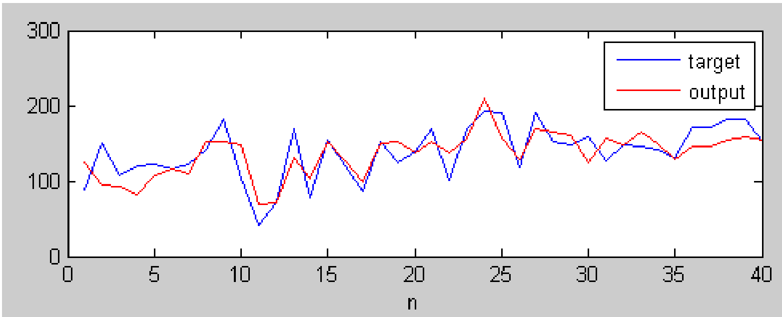


Figure 11. Prediction result using ANFIS on testing data

From the simulation, we obtain the prediction with the root of mean square error (RMSE) are :
Training data : 28.17
Testing data : 23.76

We repeat the simulations using different membership functions and the results are in table 2.

Table 2. The root of mean square error (RMSE) on ANFIS in different membership function

Membership function	Max. epoch	RMSE on training data	RMSE on testing data
gaussmf	30	28.36	27.10
trapmf	30	28.26	41.25
trimf	30	27.58	32.48
pimf	30	28.32	30.38
dsigmf	30	28.27	27.04

5. Conclusion

Both NN and ANFIS can make prediction approaching target. Simulations are applied by splitting dataset into training data (80%) and testing data (20%). From five trials of NN simulations, the number of maximum epoch for making RMSE in training data is very large so that the computation time is very long. The average RMSE in training data is 30.564. From the comparison result of two methods, we can see that ANFIS can make faster prediction than NN with the number of maximum epochs is smaller than NN so that computation time is faster.

The developments of this research are making prediction based on more than two input variables and classifying the results by data mining technique.

References

[1] Shing J, JangR 1993 ANFIS : Adaptive Network Based Fuzzy Inference System, *IEEE Transaction on System, Man and Cybernetics* **23** pp. 665-685
[2] SupartaW and Alhasa KM 2016 Modelling of Tropospheric Delays Using ANFIS*SpringerBriefs in Meteorology*
[3] Han J, Kamber M and Pei J 2012 *Data Mining Concepts and Techniques* (USA : Elsevier)
[4] Fausett L 1994*Fundamentals of Neural Networks* (USA: Prentice Hall)
[5] RahmaliaD 2018 Estimation of Exponential Smoothing Parameter on Pesticide Characteristic Forecast using Ant Colony Optimization (ACO) *Eksakta : Jurnal Ilmu-ilmu MIPA* **18** pp. 56–63.
[6] Herlambang T, Mufarrikoh Z, Karya D F and Rahmalia D 2018 Estimation of Water Level and Steam Temperature Using Ensemble Kalman Filter Square Root (EnKF-SR)*Journal of Physics : Conference Series* **1008**
[7] Herlambang T, Rasyid R A, Hartatik S and Rahmalia D 2017 Estimasi Posisi Mobile Robot Menggunakan Akar Kuadrat Unscented Kalman Filter (AK-UKF)*Technology Science and Engineering Journal* **1**
[8] Rahmalia D and Herlambang T 2018 Application Kohonen Network and Fuzzy C Means for Clustering Airports Based on Frequency of Flight*Kinetik : Game Technology, Information System, Computer Network, Computing* **3** pp. 229-236
[9] Rahmalia D and Aini N 2018 Pengaruh Korelasi Data pada Peramalan Suhu Udara Menggunakan Backpropagation Neural Network*Zeta-Math Journal* **4** pp. 1-6
[10] Rahmalia D and Herlambang T 2017 Prediksi Cuaca Menggunakan Algoritma Particle Swarm Optimization-Neural Network (PSONN) *Prosiding Seminar Nasional Matematika dan Aplikasinya* pp. 41-48
[11] Rahmalia D and Pradana M S 2019 Backpropagation Neural Network pada Data yang tak Stasioner (Studi Kasus: Jumlah Penderita Penyakit Ebola)*Jurnal Riset dan Aplikasi Matematika (JRAM)* **3**pp. 32-42
[12] Dewi C and Muslikh M 2013 Perbandingan Akurasi Backpropagation Neural Network dan ANFIS untuk Memprediksi Cuaca *Natural-A-Journal of Scientific Modeling and Computation* **1**
[13] Hidayatullah N U and Ya'umar 2014 Estimasi Radiasi Matahari Perjam pada Permukaan Horizontal dengan Learning Machine (Studi Kasus di Surabaya) *Jurnal Teknik Pomits* **2**